```
FFFFFFFFFFFFFFF        111          111      XXX         XXX
FFFFFFFFFFFFFFF        111          111      XXX         XXX
FFFFFFFFFFFFFFF        111          111      XXX         XXX
FFF                 111111       111111      XXX         XXX
FFF                 111111       111111      XXX         XXX
FFF                 111111       111111      XXX         XXX
FFF                    111          111         XXX   XXX
FFF                    111          111         XXX   XXX
FFF                    111          111         XXX   XXX
FFFFFFFF.FFF           111          111            XXX
FFFFFFFFFFFF           111          111            XXX
FFFFFFFFFFFF           111          111            XXX
FFF                    111          111         XXX   XXX
FFF                    111          111         XXX   XXX
FFF                    111          111         XXX   XXX
FFF                    111          111      XXX         XXX
FFF                    111          111      XXX         XXX
FFF                    111          111      XXX         XXX
FFF                 111111111    111111111   XXX         XXX
FFF                 111111111    111111111   XXX         XXX
FFF                 111111111    111111111   XXX         XXX
```

```
_$25
Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP_
MARK
MARK
MARK
MARK
MARK
```

```
RRRRRRRR    WW      WW    AAAAAA    TTTTTTTTTT  TTTTTTTTTT  RRRRRRRR
RRRRRRRR    WW      WW    AAAAAA    TTTTTTTTTT  TTTTTTTTTT  RRRRRRRR
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR
RRRRRRRR    WW      WW   AA    AA       TT          TT      RRRRRRRR
RRRRRRRR    WW      WW   AA    AA       TT          TT      RRRRRRRR
RR  RR      WW  WW  WW   AAAAAAAAAA       TT          TT      RR  RR
RR  RR      WW  WW  WW   AAAAAAAAAA       TT          TT      RR  RR
RR    RR    WWWW  WWWW   AA    AA       TT          TT      RR    RR      ....
RR    RR    WWWW  WWWW   AA    AA       TT          TT      RR    RR      ....
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR    ....
RR      RR  WW      WW   AA    AA       TT          TT      RR      RR    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE RWATTR (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004                    ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module contains the code and tables to process the read
;    38          0038  1 !       and write attributes functions.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines. Th s routine must be executed
;    44          0044  1 !       in kernel mode.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  6-Jan-1977  21:05
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-042 CDS0011         Christian D. Saether    30-Aug-1984
;    54          0054  1 !               Change biasing of refcnt on primary_fcb during fid_to_spec
;    55          0055  1 !               so that err_cleanup can fix if it necessary.  Also change
;    56          0056  1 !               test for removing bias at end of routine.
;    57          0057  1 !
```

```
;    58          0058  1 ;        V03-041 LMP0303         L. Mark Pilant,         21-Aug-1984  9:41
;    59          0059  1 ;                Fix a bug that caused memory to get overwritten when doing
;    60          0060  1 ;                a FID to file spec with long directory names.
;    61          0061  1 ;
;    62          0062  1 ;        V03-040 CDS0010         Christian D. Saether    16-Aug-1984
;    63          0063  1 ;                Set CLF_MARKFCBSTALE flag when modifying protected
;    64          0064  1 ;                attributes.
;    65          0065  1 ;
;    66          0066  1 ;        V03-039 CDS0009         Christian D. Saether    7-Aug-1984
;    67          0067  1 ;                Clean out directory index when turning off directory
;    68          0068  1 ;                flag in the header.
;    69          0069  1 ;
;    70          0070  1 ;        V03-038 LMP0295         L. Mark Pilant,         6-Aug-1984  16:52
;    71          0071  1 ;                Correctly return the binary version number for files with
;    72          0072  1 ;                long names.
;    73          0073  1 ;
;    74          0074  1 ;        V03-037 LMP0285         L. Mark Pilant,         26-Jul-1984  12:26
;    75          0075  1 ;                Fix a bug that caused a "?" to be added to the name if it
;    76          0076  1 ;                was 39 characters long.
;    77          0077  1 ;
;    78          0078  1 ;        V03-036 ACG0439         Andrew C. Goldstein,    20-Jul-1984  16:03
;    79          0079  1 ;                Make BYPASS priv allow change owner operations
;    80          0080  1 ;
;    81          0081  1 ;        V03-035 ACG0437         Andrew C. Goldstein,    13-Jul-1984  15:11
;    82          0082  1 ;                Add FCB arg to CHANGE_OWNER; remove ERR_EXIT calls from
;    83          0083  1 ;                CHANGE_OWNER and CHANGE_CLASS.
;    84          0084  1 ;
;    85          0085  1 ;        V03-034 LMP0259         L. Mark Pilant,         25-Jun-1984  11:39
;    86          0086  1 ;                If the user requests the matching ACE to be returned,
;    87          0087  1 ;                clear out the unused portion of the ACE.
;    88          0088  1 ;
;    89          0089  1 ;        V03-033 CDS0008         Christian D. Saether    15-May-1984
;    90          0090  1 ;                CDS0007 was not quite right.  Fix it.
;    91          0091  1 ;
;    92          0092  1 ;        V03-032 CDS0007         Christian D. Saether    11-May-1984
;    93          0093  1 ;                Don't bugcheck if there is no primary_fcb in fid_to_spec.
;    94          0094  1 ;
;    95          0095  1 ;        V03-031 CDS0006         Christian D. Saether     3-May-1984
;    96          0096  1 ;                Bump refcnt in fcb when letting go of serialization
;    97          0097  1 ;                lock in fid_to_spec.
;    98          0098  1 ;
;    99          0099  1 ;        V03-030 CDS0005         Christian D. Saether    19-Apr-1984
;   100          0100  1 ;                Use REFCNT instead of ACNT where appropriate.
;   101          0101  1 ;                Use LOCK_COUNT routine to determine if file is accessed
;   102          0102  1 ;                anywhere in possible cluster.
;   103          0103  1 ;                Remove check for erroneous acl status.
;   104          0104  1 ;
;   105          0105  1 ;        V03-029 LMPBUILD        L. Mark Pilant,         16-Apr-1984  14:12
;   106          0106  1 ;                Initialize STATUS to be SS$_NORMAL in WRITE_ATTRIBUTE.
;   107          0107  1 ;
;   108          0108  1 ;        V03-028 ACG0415         Andrew C. Goldstein,    9-Apr-1984  11:02
;   109          0109  1 ;                Add call to rebuild file ACL after modifying; fix
;   110          0110  1 ;                access mode ATR probing; maximize the access mode protection
;   111          0111  1 ;                against mode of the caller; correct all error exits to
;   112          0112  1 ;                go through the cleanup code; support extended file name
;   113          0113  1 ;                in ident area.
;   114          0114  1 ;
```

```
;  115     0115  1 !       V03-027 LMP0221         L. Mark Pilant,         1-Apr-1984  15:42
;  116     0116  1 !               Support an ORB within an FCB.
;  117     0117  1 !
;  118     0118  1 !       V03-026 ACG0412         Andrew C. Goldstein,    22-Mar-1984  18:28
;  119     0119  1 !               Implement agent access mode support; add access mode to
;  120     0120  1 !               protection check call. Also make rest of global storage based.
;  121     0121  1 !
;  122     0122  1 !       V03-025 ACG0410         Andrew C. Goldstein,    22-Mar-1984  13:45
;  123     0123  1 !               Add support for access mode attribute to probe buffers;
;  124     0124  1 !               also check for file open in CHANGE_CLASS
;  125     0125  1 !
;  126     0126  1 !       V03-024 ACG0405         Andrew C. Goldstein,    16-Mar-1984  15:15
;  127     0127  1 !               Fix handling of file headers in CHANGE_OWNER;
;  128     0128  1 !               also pull obsolete handling of exception vectors and
;  129     0129  1 !               obsolete access mask attribute.
;  130     0130  1 !
;  131     0131  1 !       V03-023 LMP0203         L. Mark Pilant,         9-Mar-1984  16:18
;  132     0132  1 !               Add support for FIB$V_PROPAGATE.
;  133     0133  1 !
;  134     0134  1 !       V03-022 CDS0004         Christian D. Saether    29-Feb-1984
;  135     0135  1 !               Serialize correctly tracking up through backlinks
;  136     0136  1 !               in the FID_TO_SPEC routine.
;  137     0137  1 !               Replace FLUSH_FID with TOSS_CACHE_DATA.
;  138     0138  1 !
;  139     0139  1 !       V03-021 LMP0195         L. Mark Pilant,         27-Feb-1984  14:38
;  140     0140  1 !               Modify the way ownership changes to allow use of the
;  141     0141  1 !               requestor's rights list.  Also, tie off classification
;  142     0142  1 !               changes.
;  143     0143  1 !
;  144     0144  1 !       V03-020 CDS0003         Christian D. Saether    19-Dec-1983
;  145     0145  1 !               Use BIND_COMMON macro to reduce external
;  146     0146  1 !               COMMON declarations.
;  147     0147  1 !
;  148     0148  1 !       V03-019 CDS0002         Christian D. Saether    14-Oct-1983
;  149     0149  1 !               NO_LCKCHK is now a byte instead of a word.
;  150     0150  1 !
;  151     0151  1 !       V03-018 CDS0001         Christian D. Saether    27-Sep-1983
;  152     0152  1 !               Set flag to disable lock basis checking for now
;  153     0153  1 !               when reading headers for back links.
;  154     0154  1 !
;  155     0155  1 !       V03-017 LMP0149         L. Mark Pilant,         8-Sep-1983  13:26
;  156     0156  1 !               Correct a logic problem that caused problems during the
;  157     0157  1 !               protection check of a write attribute operation.  Also,
;  158     0158  1 !               fix up the directory spec returned for implicitly spooled
;  159     0159  1 !               files.
;  160     0160  1 !
;  161     0161  1 !       V03-016 LMP0141         L. Mark Pilant,         24-Aug-1983  3:10
;  162     0162  1 !               Return '?' for a zero back-link only if a non-zero link was
;  163     0163  1 !               previously seen.
;  164     0164  1 !
;  165     0165  1 !       V03-015 LMP0131         L. Mark Pilant,         2-Aug-1983  11:06
;  166     0166  1 !               Check for a minimum attribute length when writing the
;  167     0167  1 !               reserved area of the file header.
;  168     0168  1 !
;  169     0169  1 !       V03-014 LMP0129         L. Mark Pilant,         25-Jul-1983  11:51
;  170     0170  1 !               Return the correct directory name for files in the MFD.
;  171     0171  1 !
```

```
;  172          0172  1 ;       V03-013 LMP0120         L. Mark Pilant,         22-Jun-1983  8:41
;  173          0173  1 ;               Correct the sizes used for the ACL attributes.
;  174          0174  1 ;
;  175          0175  1 ;       V03-012 ACG0338         Andrew C. Goldstein,    1-Jun-1983  17:21
;  176          0176  1 ;               Fix attribute control table change
;  177          0177  1 ;
;  178          0178  1 ;       V03-011 LMP0113         L. Mark Pilant,         10-May-1983  13:38
;  179          0179  1 ;               Add support for FID to file-spec translation.
;  180          0180  1 ;
;  181          0181  1 ;       V03-010 LMP0110         L. Mark Pilant,         3-May-1983  12:46
;  182          0182  1 ;               Add support for access allowed, privs used, and ACE used
;  183          0183  1 ;               to gain access.
;  184          0184  1 ;
;  185          0185  1 ;       V03-009 STJ3096         Steven T. Jeffreys,     29-Apr-1983
;  186          0186  1 ;               Make FH2$M_NOCHARGE a protected file characteristic.
;  187          0187  1 ;
;  188          0188  1 ;       V03-008 LMP0105         L. Mark Pilant,         26-Apr-1983  16:18
;  189          0189  1 ;               Add table entries for several new attributes.  Also, fix a
;  190          0190  1 ;               problem with issuing read type ACL attributes during a write
;  191          0191  1 ;               attribute operation.
;  192          0192  1 ;
;  193          0193  1 ;       V03-007 ACG0329         Andrew C. Goldstein,    12-Apr-1983  13:59
;  194          0194  1 ;               Fold long UIC's into [377,377] for 16 bit UIC
;  195          0195  1 ;
;  196          0196  1 ;       V03-006 STJ3078         Steven T. Jeffreys,     26-Mar-1983
;  197          0197  1 ;               Add support for the HIGHWATER mark.
;  198          0198  1 ;
;  199          0199  1 ;       V03-005 LMP0088         L. Mark Pilant,         16-Mar-1983  15:18
;  200          0200  1 ;               Add support for reading a single ACE and reading and writing
;  201          0201  1 ;               the reserved area in the file header.
;  202          0202  1 ;
;  203          0203  1 ;       V03-004 ACG0306         Andrew C. Goldstein,    13-Dec-1982  14:36
;  204          0204  1 ;               Get rid of obsolete file structure names
;  205          0205  1 ;
;  206          0206  1 ;       V03-003 LMP0054         L. Mark Pilant,         25-Oct-1982  17:05
;  207          0207  1 ;               Add a new attribute to return the ACL length.
;  208          0208  1 ;
;  209          0209  1 ;       V03-002 LMP0045         L. Mark Pilant,         21-Sep-1982  13:10
;  210          0210  1 ;               Call the ACL dispatcher in kernel mode when writing attributes.
;  211          0211  1 ;
;  212          0212  1 ;       V03-001 LMP0036         L. Mark Pilant,         29-Jun-1982  16:25
;  213          0213  1 ;               Add support for Access Control Lists.
;  214          0214  1 ;
;  215          0215  1 ;       V02-013 ACG43657        Andrew C. Goldstein,    17-Feb-1982  19:14
;  216          0216  1 ;               Fix bug in rereading primary header after changing file owner
;  217          0217  1 ;               of a multi-header file
;  218          0218  1 ;
;  219          0219  1 ;       V02-012 ACG0253         Andrew C. Goldstein,    18-Jan-1982  16:30
;  220          0220  1 ;               Add dummy HDR1 accessibility attribute
;  221          0221  1 ;
;  222          0222  1 ;       V02-011 ACG0241         Andrew C. Goldstein,    11-Dec-1981  22:32
;  223          0223  1 ;               Force RMS directory cache flush when directory bit is cleared
;  224          0224  1 ;
;  225          0225  1 ;       V02-010 ACG0232         Andrew C. Goldstein,    4-Dec-1981  16:43
;  226          0226  1 ;               Protect HIBLK during write attributes operation
;  227          0227  1 ;
;  228          0228  1 ;       V02-009 ACG0229         Andrew C. Goldstein,    1-Dec-1981  0:38
```

```
;  229          0229  1 !                       Add full counts and I/O counters to stat block
;  230          0230  1 !
;  231          0231  1 !               V02-008 ACG0221         Andrew C. Goldstein,    30-Oct-1981  18:04
;  232          0232  1 !                       Add attribute for journal control flags
;  233          0233  1 !
;  234          0234  1 !               V02-007 ACG0196         Andrew C. Goldstein,    5-Mar-1981  16:32
;  235          0235  1 !                       Fix file header length checks
;  236          0236  1 !
;  237          0237  1 !               V02-006 ACG0190         Andrew C. Goldstein,    16-Feb-1981  11:25
;  238          0238  1 !                       Remove old security mask field
;  239          0239  1 !
;  240          0240  1 !               V02-005 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:27
;  241          0241  1 !                       Previous revision history moved to F11B.REV
;  242          0242  1 !**
;  243          0243  1
;  244          0244  1
;  245          0245  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  246          0246  1 REQUIRE 'SRC$:FCPDEF.B32';
;  247          1237  1
;  248          1238  1
;  249          1239  1 FORWARD ROUTINE
;  250          1240  1         READ_ATTRIB     : L_NORM,       ! read attributes
;  251          1241  1         FID_TO_SPEC     : L_NORM NOVALUE, ! convert FID to file-spec
;  252          1242  1         READ_HANDLER    : NOVALUE,      ! read attributes condition handler
;  253          1243  1         WRITE_ATTRIB    : L_NORM NOVALUE, ! write attributes
;  254          1244  1         CONVERT_DATE    : L_NORM NOVALUE, ! convert string date to 64 bit
;  255          1245  1         CHANGE_OWNER    : L_NORM,       ! change file owner UIC
;  256          1246  1         CHANGE_CLASS    : L_NORM;       ! Change file classification
```

```
;  258          1247  1 !++
;  259          1248  1 !
;  260          1249  1 ! Attribute control table. The table is indexed by attribute number.
;  261          1250  1 ! Each entry is a quadword.
;  262          1251  1 !
;  263          1252  1 !--
;  264          1253  1
;  265          1254  1 ! Macros and literals to access the table entries.
;  266          1255  1 !
;  267          1256  1
;  268          1257  1 MACRO
;  269          1258  1                                                 ! flags byte
;  270          1259  1         ATC_READ_ONLY   = 0,0,1,0%,             ! read only attribute
;  271          1260  1         ATC_PROTECTED   = 0,1,1,0%,             ! writable by file owner only
;  272          1261  1         ATC_LOCKED      = 0,2,1,0%,             ! subject to file access locks
;  273          1262  1         ATC_LOCATION    = 1,0,8,0%,             ! location code
;  274          1263  1         ATC_OFFSET      = 2,0,8,0%,             ! location offset
;  275          1264  1         ATC_ACTION      = 3,0,8,0%,             ! action routine
;  276          1265  1         ATC_DATA_SIZE   = 4,0,16,0%,            ! size of data area holding attribute
;  277          1266  1         ATC_MAX_SIZE    = 6,0,16,0%;            ! max attribute size
;  278          1267  1
;  279          1268  1
;  280          1269  1 ! Masks for the flags.
;  281          1270  1 !
;  282          1271  1
;  283          1272  1 LITERAL
;  284          1273  1         M_READ_ONLY     = 1,
;  285          1274  1         M_PROTECTED     = 2,
;  286          1275  1         M_LOCKED        = 4;
;  287          1276  1
;  288          1277  1
;  289          1278  1 ! Attribute location codes.
;  290          1279  1 !
;  291          1280  1
;  292          1281  1 LITERAL
;  293          1282  1         ATC_ZERO        = 0,            ! zero - no location
;  294          1283  1         ATC_FCB         = 1,            ! in file control block
;  295          1284  1         ATC_HEADER      = 2,            ! file header header area
;  296          1285  1         ATC_IDENT       = 3,            ! file header ident area
;  297          1286  1         ATC_MAP         = 4,            ! file header map area
;  298          1287  1         ATC_ACL         = 5,            ! file header Access Control List area
;  299          1288  1         ATC_RESERVED    = 6,            ! file header reserved area
;  300          1289  1         ATC_ACPGBL      = 7,            ! ACP global storage
;  301          1290  1         ATC_FID2NAME    = 8,            ! Convert FID to file spec
;  302          1291  1
;  303          1292  1         ATC_LASTATC     = 8;            ! last location code
;  304          1293  1
;  305          1294  1 ! ACP global storage index codes.
;  306          1295  1
;  307          1296  1 LITERAL
;  308          1297  1         GBL_PRV         = 0,            ! Privileges used to gain access
;  309          1298  1         GBL_ACE         = 1,            ! ACE used to gain access
;  310          1299  1
;  311          1300  1         GBL_LASTGBL     = 1;            ! Last index code
;  312          1301  1
;  313          1302  1
;  314          1303  1 ! Attribute processing action routines.
```

```
;  315     1304  1 !
;  316     1305  1
;  317     1306  1 LITERAL
;  318     1307  1         ACT_NOP         = 0,        ! ignore attribute
;  319     1308  1         ACT_ILLEGAL     = 1,        ! illegal attribute code
;  320     1309  1         ACT_COPY        = 2,        ! simple copy
;  321     1310  1         ACT_STATBLK     = 3,        ! build statistics block
;  322     1311  1         ACT_ZERO        = 4,        ! zero valued attribute
;  323     1312  1         ACT_BLOCKSIZE   = 5,        ! medium block size
;  324     1313  1         ACT_R50_NAME    = 6,        ! RAD-50 file name, type, version
;  325     1314  1         ACT_R50_TYPE    = 7,        ! RAD-50 file type & version
;  326     1315  1         ACT_R50_VER     = 8,        ! binary version number
;  327     1316  1         ACT_UIC2        = 9,        ! 2 byte file owner, protection, char
;  328     1317  1         ACT_FPRO        = 10,       ! file protection + characteristics
;  329     1318  1         ACT_DATE        = 11,       ! ASCII date
;  330     1319  1         ACT_DATES       = 12,       ! revision count, ASCII dates
;  331     1320  1         ACT_UIC4        = 13,       ! 4 byte file owner UIC
;  332     1321  1         ACT_BLANK       = 14,       ! blank values attribute
;  333     1322  1         ACT_ACL         = 15,       ! Access Control List
;  334     1323  1         ACT_RESERVED    = 16,       ! Reserved area
;  335     1324  1         ACT_CLASS       = 17,       ! Classification mask
;  336     1325  1         ACT_ACMODE      = 18,       ! buffer access mode
;  337     1326  1         ACT_ACLEVEL     = 19,       ! access mode protection of file
;  338     1327  1         ACT_FILENAME    = 20,       ! internal file name
;  339     1328  1
;  340     1329  1         ACT_LASTACT     = 20;       ! highest action routine code
;  341     1330  1
;  342     1331  1
;  343     1332  1 ! Macro to build table entry.
;  344     1333  1 !
;  345     1334  1
;  346     1335  1 MACRO
;  347   M 1336  1         ATTRIBUTE (CODE, FLAGS, LOC, OFF1, OFF2, OFF3, OFF4, SIZE, DATA, ACTION) =
;  348   M 1337  1                 BYTE    (FLAGS,
;  349   M 1338  1                          LOC,
;  350   M 1339  1                          $BYTEOFFSET (OFF1, OFF2, OFF3, OFF4),
;  351   M 1340  1                          ACTION),
;  352   M 1341  1                 WORD    (DATA,
;  353   M 1342  1                          SIZE)
;  354     1343  1                 %;
;  355     1344  1
;  356     1345  1 MACRO
;  357     1346  1         NULL_FIELD      = 0,0,0,0%;
;  358     1347  1
;  359     1348  1
;  360     1349  1 ! The attribute control table itself.
;  361     1350  1 !
;  362     1351  1
;  363     1352  1 BIND
;  364     1353  1         ATC     = UPLIT (
;  365     1354  1
;  366     1355  1 ATTRIBUTE (0,                  M_PROTECTED, ATC_HEADER,   FH2$L_FILEOWNER,  5,  6, ACT_UIC2),
;  367     1356  1 ATTRIBUTE (0,                  M_PROTECTED, ATC_HEADER,   FH2$W_FILEPROT,   3,  2, ACT_FPRO),
;  368     1357  1 ATTRIBUTE (ATR$C_UCHAR,        M_LOCKED,    ATC_HEADER,   FH2$L_FILECHAR,  4,  4, ACT_COPY),
;  369     1358  1 ATTRIBUTE (ATR$C_RECATTR,      M_LOCKED,    ATC_HEADER,   FH2$W_RECATTR,  32, 32, ACT_COPY),
;  370     1359  1 ATTRIBUTE (ATR$C_FILNAM,       0,           ATC_IDENT,    FI2$T_FILENAME, 10, 20, ACT_R50_NAME),
;  371     1360  1 ATTRIBUTE (ATR$C_FILTYP,       0,           ATC_IDENT,    FI2$T_FILENAME,  4, 20, ACT_R50_TYPE),
```

```
;  372          1361  1 ATTRIBUTE (ATR$C_FILVER,      0,           ATC_IDENT,    FI2$T_FILENAME,    2, 20, ACT_R50_VER),
;  373          1362  1 ATTRIBUTE (ATR$C_EXPDAT,      M_PROTECTED, ATC_IDENT,    FI2$Q_EXPDATE,     7,  8, ACT_DATE),
;  374          1363  1 ATTRIBUTE (ATR$C_STATBLK,     M_READ_ONLY, ATC_FCB,      NULL_FIELD,       32,  0, ACT_STATBLK),
;  375          1364  1 ATTRIBUTE (ATR$C_HEADER,      M_READ_ONLY, ATC_HEADER,   NULL_FIELD,      512,  0, ACT_COPY),
;  376          1365  1 ATTRIBUTE (ATR$C_BLOCKSIZE,   0,           ATC_ZERO,     NULL_FIELD,        2,  0, ACT_BLOCKSIZE),
;  377          1366  1 ATTRIBUTE (ATR$C_USERLABEL,   0,           ATC_ZERO,     NULL_FIELD,       80, 80, ACT_ZERO),
;  378          1367  1 ATTRIBUTE (ATR$C_ASCDATES,    M_PROTECTED, ATC_IDENT,    NULL_FIELD,       35, 46, ACT_DATES),
;  379          1368  1 ATTRIBUTE (ATR$C_ALCONTROL,   0,           ATC_ZERO,     NULL_FIELD,       14,  0, ACT_NOP),
;  380          1369  1 ATTRIBUTE (ATR$C_ENDLBLAST,   0,           ATC_ZERO,     NULL_FIELD,        1,  0, ACT_NOP),
;  381          1370  1 ATTRIBUTE (ATR$C_ASCNAME,     0,           ATC_IDENT,    NULL_FIELD,       86, 20, ACT_FILENAME),
;  382          1371  1 ATTRIBUTE (ATR$C_CREDATE,     M_PROTECTED, ATC_IDENT,    FI2$Q_CREDATE,     8,  8, ACT_COPY),
;  383          1372  1 ATTRIBUTE (ATR$C_REVDATE,     M_PROTECTED, ATC_IDENT,    FI2$Q_REVDATE,     8,  8, ACT_COPY),
;  384          1373  1 ATTRIBUTE (ATR$C_EXPDATE,     M_PROTECTED, ATC_IDENT,    FI2$Q_EXPDATE,     8,  8, ACT_COPY),
;  385          1374  1 ATTRIBUTE (ATR$C_BAKDATE,     M_PROTECTED, ATC_IDENT,    FI2$Q_BAKDATE,     8,  8, ACT_COPY),
;  386          1375  1 ATTRIBUTE (ATR$C_UIC,         M_PROTECTED, ATC_HEADER,   FH2$L_FILEOWNER,   4,  4, ACT_UIC4),
;  387          1376  1 ATTRIBUTE (ATR$C_FPRO,        M_PROTECTED, ATC_HEADER,   FH2$W_FILEPROT,    2,  2, ACT_COPY),
;  388          1377  1 ATTRIBUTE (ATR$C_RPRO,        M_PROTECTED, ATC_HEADER,   FH2$W_RECPROT,     2,  2, ACT_COPY),
;  389          1378  1 ATTRIBUTE (ATR$C_ACLEVEL,     M_PROTECTED, ATC_HEADER,   FH2$B_ACC_MODE,    1,  1, ACT_ACLEVEL),
;  390          1379  1 ATTRIBUTE (ATR$C_SEMASK,      M_PROTECTED, ATC_HEADER,   NULL_FIELD,        8,  8, ACT_ZERO),
;  391          1380  1 ATTRIBUTE (ATR$C_UIC_RO,      M_READ_ONLY, ATC_HEADER,   FH2$L_FILEOWNER,   4,  4, ACT_COPY),
;  392          1381  1 ATTRIBUTE (ATR$C_DIRSEQ,      M_READ_ONLY, ATC_FCB,      FCB$W_DIRSEQ,      2,  0, ACT_COPY),
;  393          1382  1 ATTRIBUTE (ATR$C_BACKLINK,    M_PROTECTED, ATC_HEADER,   FH2$W_BACKLINK,    6,  6, ACT_COPY),
;  394          1383  1 ATTRIBUTE (ATR$C_JOURNAL,     M_PROTECTED, ATC_HEADER,   FH2$W_JOURNAL,     2,  2, ACT_COPY),
;  395          1384  1 ATTRIBUTE (ATR$C_HDR1_ACC,    M_PROTECTED, ATC_ZERO,     NULL_FIELD,        1,  0, ACT_BLANK),
;  396          1385  1 ATTRIBUTE (ATR$C_ADDACLENT,   M_LOCKED,    ATC_ACL,      NULL_FIELD,      512,  0, ACT_ACL),
;  397          1386  1 ATTRIBUTE (ATR$C_DELACLENT,   M_LOCKED,    ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  398          1387  1 ATTRIBUTE (ATR$C_MODACLENT,   M_LOCKED,    ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  399          1388  1 ATTRIBUTE (ATR$C_FNDACLENT,   M_READ_ONLY, ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  400          1389  1 ATTRIBUTE (ATR$C_FNDACETYP,   M_READ_ONLY, ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  401          1390  1 ATTRIBUTE (ATR$C_DELETEACL,   M_LOCKED,    ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  402          1391  1 ATTRIBUTE (ATR$C_READACL,     M_READ_ONLY, ATC_ACL,      NULL_FIELD,      512,  0, ACT_ACL),
;  403          1392  1 ATTRIBUTE (ATR$C_ACLLENGTH,   M_READ_ONLY, ATC_ACL,      NULL_FIELD,        4,  0, ACT_ACL),
;  404          1393  1 ATTRIBUTE (ATR$C_READACE,     M_READ_ONLY, ATC_ACL,      NULL_FIELD,      255,  0, ACT_ACL),
;  405          1394  1 ATTRIBUTE (ATR$C_RESERVED,    M_PROTECTED, ATC_RESERVED, NULL_FIELD,      380,  0, ACT_RESERVED),
;  406          1395  1 ATTRIBUTE (ATR$C_HIGHWATER,   M_READ_ONLY, ATC_HEADER,   FH2$L_HIGHWATER,  4,  4, ACT_COPY),
;  407          1396  1 ATTRIBUTE (ATR$C_ACCESS_MASK, M_READ_ONLY, ATC_ZERO,     NULL_FIELD,        0,  0, ACT_ILLEGAL),
;  408          1397  1 ATTRIBUTE (ATR$C_PRIVS_USED,  M_READ_ONLY, ATC_ACPGBL,   GBL_PRV,0,0,0,    4,  0, ACT_COPY),
;  409          1398  1 ATTRIBUTE (ATR$C_MATCHING_ACE,M_READ_ONLY, ATC_ACPGBL,   GBL_ACE,0,0,0,  255,  0, ACT_COPY),
;  410          1399  1 ATTRIBUTE (ATR$C_ACCESS_MODE, M_READ_ONLY, ATC_ZERO,     NULL_FIELD,        1,  0, ACT_ACMODE),
;  411          1400  1 ATTRIBUTE (ATR$C_FILE_SPEC,   M_READ_ONLY, ATC_FID2NAME, NULL_FIELD,      512,  0, ACT_COPY),
;  412        P 1401  1 ATTRIBUTE (ATR$C_CLASS_MASK,  M_PROTECTED OR
;  413          1402  1                              M_LOCKED,    ATC_HEADER,   FH2$R_CLASS_PROT,20, 20, ACT_CLASS),
;  414          1403  1 ATTRIBUTE (ATR$C_BUFFER_OFFSET, 0,         ATC_ZERO,     NULL_FIELD,        0,  0, ACT_NOP)
;  415          1404  1
;  416          1405  1                 ) : BBLOCKVECTOR [,8];
;  417          1406  1
;  418          1407  1 LITERAL
;  419          1408  1         MAX_CODE        = ATR$C_MAX_CODE;       ! highest attribute code
;  420          1409  1
;  421          1410  1
;  422          1411  1 !
;  423          1412  1 ! Protected bits in the file characteristics longword. These may not be
;  424          1413  1 ! modified by write attributes calls.
;  425          1414  1 !
;  426          1415  1 LITERAL
;  427          1416  1         PROTECTED_CHAR  =  FH2$M_CONTIG
;  428          1417  1                         OR FH2$M_SPOOL
```

```
;  429          1418  1                    OR FH2$M_BADBLOCK
;  430          1419  1                    OR FH2$M_NOCHARGE
;  431          1420  1                    OR FH2$M_MARKDEL;
```

```
;  433     1421  1 GLOBAL ROUTINE READ_ATTRIB (HEADER, ABD) : L_NORM =
;  434     1422  1
;  435     1423  1 !++
;  436     1424  1 !
;  437     1425  1 ! FUNCTIONAL DESCRIPTION:
;  438     1426  1 !
;  439     1427  1 !       This routine performs the read attributes function. The
;  440     1428  1 !       requested attributes are assembled into the buffer packet.
;  441     1429  1 !
;  442     1430  1 ! CALLING SEQUENCE:
;  443     1431  1 !       READ_ATTRIB (ARG1, ARG2)
;  444     1432  1 !
;  445     1433  1 ! INPUT PARAMETERS:
;  446     1434  1 !       ARG1: address of file header
;  447     1435  1 !       ARG2: address of buffer descriptors
;  448     1436  1 !
;  449     1437  1 ! IMPLICIT INPUTS:
;  450     1438  1 !       IO_PACKET: I/O packet for this operation
;  451     1439  1 !       PRIMARY_FCB: FCB of file
;  452     1440  1 !
;  453     1441  1 ! OUTPUT PARAMETERS:
;  454     1442  1 !       ARG2: address of buffer descriptors
;  455     1443  1 !
;  456     1444  1 ! IMPLICIT OUTPUTS:
;  457     1445  1 !       NONE
;  458     1446  1 !
;  459     1447  1 ! ROUTINE VALUE:
;  460     1448  1 !       1 if successful
;  461     1449  1 !       0 if error
;  462     1450  1 !
;  463     1451  1 ! SIDE EFFECTS:
;  464     1452  1 !       attribute data written into buffer packet
;  465     1453  1 !
;  466     1454  1 !--
;  467     1455  1
;  468     1456  2 BEGIN
;  469     1457  2
;  470     1458  2 MAP
;  471     1459  2         HEADER          : REF BBLOCK,   ! file header arg
;  472     1460  2         ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH];
;  473     1461  2                                         ! buffer descriptor arg
;  474     1462  2
;  475     1463  2 LOCAL
;  476     1464  2         LOCAL_HEADER    : REF BBLOCK,   ! local copy of the header address
;  477     1465  2         ACCESS_MODE,                    ! access mode to set for attribute buffer
;  478     1466  2         STATUS;                         ! routine exit status
;  479     1467  2
;  480     1468  2 BIND_COMMON;
;  481     1469  2
;  482     1470  2 EXTERNAL ROUTINE
;  483     1471  2         PMS_START_SUB   : L_NORM,       ! start subfunction metering
;  484     1472  2         PMS_END_SUB     : L_NORM,       ! end subfunction metering
;  485     1473  2         GET_TIME        : L_NORM,       ! convert 64 bit time to ASCII
;  486     1474  2         MAKE_NAMEBLOCK  : L_NORM,       ! convert file string into RAD-50
;  487     1475  2         ACL_DISPATCH    : L_NORM;       ! ACL action dispatcher
;  488     1476  2
;  489     1477  2
```

```
;  490          1478  2 ENABLE READ_HANDLER;
;  491          1479  2
;  492          1480  2 ! Start metering for this subfunction.
;  493          1481  2 !
;  494          1482  2
;  495          1483  2 PMS_START_SUB (PMS_RWATT);
;  496          1484  2
;  497          1485  2 STATUS = 1;                              ! assume success
;  498          1486  2 LOCAL_HEADER = .HEADER;                  ! Copy header address
;  499          1487  2
;  500          1488  2 ! Set the buffered read bit in the I/O packet to indicate to IO_DONE that
;  501          1489  2 ! the attribute buffers are valid.
;  502          1490  2 !
;  503          1491  2
;  504          1492  2 IO_PACKET[IRP$V_FUNC] = 1;
;  505          1493  2 ACCESS_MODE = .IO_PACKET[IRP$V_MODE];
;  506          1494  2
;  507          1495  2 ! Scan the buffer packet, picking up each entry. The first byte of the
;  508          1496  2 ! text is the attribute code, and must be overwritten with the access
;  509          1497  2 ! mode of the request for the I/O completion processing.
;  510          1498  2 !
;  511          1499  2
;  512          1500  2 INCR I FROM ABD$C_ATTRIB TO .IO_PACKET[IRP$W_BCNT]-1 DO
;  513          1501  3     BEGIN
;  514          1502  3
;  515          1503  3     LITERAL
;  516          1504  3         ATB_LENGTH      = FI2$S_FILENAME+FI2$S_FILENAMEXT; ! length of temp attribute buffer
;  517          1505  3
;  518          1506  3     LOCAL
;  519          1507  3         P,                                ! pointer to attribute text
;  520          1508  3         T,                                ! temporary pointer
;  521          1509  3         COUNT,                            ! attribute size desired
;  522          1510  3         ADDRESS         : REF BBLOCK,     ! address of attribute
;  523          1511  3         CODE,                             ! attribute code
;  524          1512  3         MAX_COUNT,                        ! max size of attribute
;  525          1513  3         ACTION          : BYTE,           ! code of action routine
;  526          1514  3         NAME_BLOCK      : BBLOCK [NMB$C_LENGTH], ! buffer for file name block
;  527          1515  3         ATT_BUFFER      : BBLOCK [ATB_LENGTH];  ! buffer to build reformatted attribute text
;  528          1516  3
;  529          1517  3     P = .ABD[.I, ABD$W_TEXT] + ABD[.I, ABD$W_TEXT];
;  530          1518  3     COUNT = .ABD[.I, ABD$W_COUNT];
;  531          1519  3     CODE = .(.P)<0,8> - 1;
;  532          1520  3     (.P)<0,8> = .ACCESS_MODE;
;  533          1521  3     P = .P + 1;
;  534          1522  3
;  535          1523  3 ! Check the attribute code for legality, and then check the requested
;  536          1524  3 ! size against the limit. If an error exit is made, first truncate the
;  537          1525  3 ! descriptor count to inhibit return of the unprocessed descriptors.
;  538          1526  3 !
;  539          1527  3
;  540          1528  3     IF .CODE GTR MAX_CODE - 1
;  541          1529  3     THEN
;  542          1530  4         BEGIN
;  543          1531  4         IO_PACKET[IRP$W_BCNT] = .I;
;  544          1532  4         (ERR_STATUS (SS$_BADATTRIB); STATUS = 0; EXITLOOP);
;  545          1533  3         END;
;  546          1534  3
```

```
;  547          1535  3     MAX_COUNT = .ATC[.CODE, ATC_MAX_SIZE];
;  548          1536  3     IF .COUNT GTR .MAX_COUNT
;  549          1537  3     THEN
;  550          1538  4         BEGIN
;  551          1539  4         IO_PACKET[IRP$W_BCNT] = .I;
;  552          1540  4         (ERR_STATUS (SS$_BADATTRIB); STATUS = 0; EXITLOOP);
;  553          1541  3         END;
;  554          1542  3
;  555          1543  3
;  556          1544  3 ! Get the action routine code first.
;  557          1545  3 !
;  558          1546  3
;  559          1547  3     ACTION = .ATC[.CODE, ATC_ACTION];
;  560          1548  3
;  561          1549  3 ! Compute the address of the attribute.
;  562          1550  3 !
;  563          1551  3
;  564          1552  3     ADDRESS =
;  565          1553  4         (
;  566          1554  4         CASE .ATC[.CODE, ATC_LOCATION] FROM 0 TO ATC_LASTATC OF
;  567          1555  4             SET
;  568          1556  4             [ATC_ZERO]:         ATT_BUFFER;
;  569          1557  4             [ATC_FCB]:          .PRIMARY_FCB;
;  570          1558  5             [ATC_HEADER]:       BEGIN
;  571          1559  5                                 IF .ATC[.CODE, ATC_OFFSET]
;  572          1560  5                                  + .ATC[.CODE, ATC_DATA_SIZE] GTRU
;  573          1561  5                                    .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  574          1562  5                                 THEN ACTION = ACT_ZERO;
;  575          1563  5                                 .LOCAL_HEADER
;  576          1564  4                                 END;
;  577          1565  5             [ATC_IDENT]:        BEGIN
;  578          1566  5                                 IF .ATC[.CODE, ATC_OFFSET]
;  579          1567  5                                  + .ATC[.CODE, ATC_DATA_SIZE] GTRU
;  580          1568  5                                    .LOCAL_HEADER[FH2$B_MPOFFSET]*2
;  581          1569  5                                  - .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  582          1570  5                                 THEN ACTION = ACT_ZERO;
;  583          1571  5                                 .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  584          1572  4                                 END;
;  585          1573  4             [ATC_MAP]:          .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_MPOFFSET]*2;
;  586          1574  4             [ATC_ACL]:          .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_ACOFFSET]*2;
;  587          1575  4             [ATC_RESERVED]:     .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_RSOFFSET]*2;
;  588          1576  5             [ATC_ACPGBL]:       (CASE .ATC[.CODE, ATC_OFFSET] FROM 0 TO GBL_LASTGBL OF
;  589          1577  5                                     SET
;  590          1578  5                                     [GBL_PRV]:  PRIVS_USED;
;  591          1579  6                                     [GBL_ACE]:  BEGIN
;  592          1580  6                                                 CH$FILL (0, ATR$S_READACE - .MATCHING_ACE[ACE$B_SIZE],
;  593          1581  6                                                             MATCHING_ACE + .MATCHING_ACE[ACE$B_SIZE]);
;  594          1582  6                                                 MATCHING_ACE
;  595          1583  5                                                 END;
;  596          1584  4                                     TES) - .ATC[.CODE, ATC_OFFSET];
;  597          1585  5             [ATC_FID2NAME]:     BEGIN
;  598          1586  5                                 FID_TO_SPEC (.LOCAL_HEADER);
;  599          1587  5                                 LOCAL_HEADER = .FILE_HEADER;
;  600          1588  5                                 FILE_SPEC_LEN
;  601          1589  4                                 END;
;  602          1590  4             TES
;  603          1591  4         )
```

```
;  604          1592  3          + .ATC[.CODE, ATC_OFFSET];
;  605          1593  3
;  606          1594  3
;  607          1595  3      ! Finally execute the action routine.
;  608          1596  3      !
;  609          1597  3
;  610          1598  3          CASE .ACTION FROM 0 TO ACT_LASTACT OF
;  611          1599  3              SET
;  612          1600  3
;  613          1601  3              [ACT_NOP]:      COUNT = 0;
;  614          1602  3
;  615          1603  4              [ACT_ILLEGAL]:  BEGIN
;  616          1604  4                              IO_PACKET[IRP$W_BCNT] = .I;
;  617          1605  4                              (ERR_STATUS (SS$_BADATTRIB); STATUS = 0; EXITLOOP);
;  618          1606  3                              END;
;  619          1607  3
;  620          1608  3              [ACT_COPY,
;  621          1609  3               ACT_ACLEVEL,
;  622          1610  3               ACT_UIC[4]]:   0;
;  623          1611  3
;  624          1612  4              [ACT_CLASS]:    BEGIN
;  625          1613  4                              CH$FILL (0, .COUNT, .P);
;  626          1614  4                              IF .HEADER[FH2$B_IDOFFSET]*2 LEQU FH2$C_LENGTH
;  627          1615  4                              THEN COUNT = 0;
;  628          1616  3                              END;
;  629          1617  3
;  630          1618  4              [ACT_STATBLK]:  BEGIN
;  631          1619  4                              ATT_BUFFER[SBK$L_STLBN] = ROT (.ADDRESS[FCB$L_STLBN], 16);
;  632          1620  4                              ATT_BUFFER[SBK$L_FILESIZE] = ROT (.ADDRESS[FCB$L_FILESIZE], 16);
;  633          1621  4                              ATT_BUFFER[SBK$B_ACNT]  = .ADDRESS[FCB$W_ACNT];
;  634          1622  4                              ATT_BUFFER[SBK$B_LCNT]  = .ADDRESS[FCB$W_LCNT];
;  635          1623  4                              ATT_BUFFER[SBK$L_FCB] = .ADDRESS;
;  636          1624  4                              (ATT_BUFFER[SBK$L_FCB]+4)<0,16> = 0; ! unused field
;  637          1625  4                              ATT_BUFFER[SBK$W_ACNT] = .ADDRESS[FCB$W_ACNT];
;  638          1626  4                              ATT_BUFFER[SBK$W_LCNT] = .ADDRESS[FCB$W_LCNT];
;  639          1627  4                              ATT_BUFFER[SBK$W_WCNT] = .ADDRESS[FCB$W_WCNT];
;  640          1628  4                              ATT_BUFFER[SBK$W_TCNT] = .ADDRESS[FCB$W_TCNT];
;  641          1629  4                              ATT_BUFFER[SBK$L_READS] = 0;
;  642          1630  4                              ATT_BUFFER[SBK$L_WRITES] = 0;
;  643          1631  4                              IF .CURRENT_WINDOW NEQ 0
;  644          1632  4                              THEN
;  645          1633  5                                  BEGIN
;  646          1634  5                                  ATT_BUFFER[SBK$L_READS] = .CURRENT_WINDOW[WCB$L_READS];
;  647          1635  5                                  ATT_BUFFER[SBK$L_WRITES] = .CURRENT_WINDOW[WCB$L_WRITES];
;  648          1636  4                                  END;
;  649          1637  4                              ADDRESS = ATT_BUFFER;
;  650          1638  3                              END;
;  651          1639  3
;  652          1640  3          [ACT_BLOCKSIZE]: ADDRESS = UPLIT (512);
;  653          1641  3
;  654          1642  4          [ACT_ZERO]:     BEGIN
;  655          1643  4                          CH$FILL (0, .COUNT, .P);
;  656          1644  4                          COUNT = 0;
;  657          1645  3                          END;
;  658          1646  3
;  659          1647  4          [ACT_BLANK]:    BEGIN
;  660          1648  4                          ADDRESS = UPLIT BYTE (' ');
```

```
;  661          1649  3                          END;
;  662          1650  3
;  663          1651  4          [ACT_UIC2]:     BEGIN
;  664          1652  4                          MAP ATT_BUFFER : VECTOR [,BYTE];
;  665          1653  4                          ATT_BUFFER[0] = .(.ADDRESS)<0,8>;
;  666          1654  4                          ATT_BUFFER[1] = .(.ADDRESS)<16,8>;
;  667          1655  4                          IF .(.ADDRESS)<8,8> NEQ 0
;  668          1656  4                          OR .(.ADDRESS)<24,8> NEQ 0
;  669          1657  4                          THEN
;  670          1658  5                              BEGIN
;  671          1659  5                              ATT_BUFFER[0] = -1;
;  672          1660  5                              ATT_BUFFER[1] = -1;
;  673          1661  4                              END;
;  674          1662  4                          (ATT_BUFFER[2])<0,16> = .LOCAL_HEADER[FH2$W_FILEPROT];
;  675          1663  4                          ATT_BUFFER[4] = .LOCAL_HEADER[FH2$L_FILECHAR];
;  676          1664  4                          ADDRESS = ATT_BUFFER;
;  677          1665  3                          END;
;  678          1666  3
;  679          1667  4          [ACT_FPRO]:     BEGIN
;  680          1668  4                          MAP ATT_BUFFER : VECTOR [,BYTE];
;  681          1669  4                          (ATT_BUFFER[0])<0,16> = .(.ADDRESS)<0,16>;
;  682          1670  4                          ATT_BUFFER[2] = .LOCAL_HEADER[FH2$L_FILECHAR];
;  683          1671  4                          ADDRESS = ATT_BUFFER;
;  684          1672  3                          END;
;  685          1673  3
;  686          1674  4          [ACT_FILENAME]: BEGIN
;  687          1675  4                          T = ATT_BUFFER;
;  688          1676  4                          CH$COPY (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ' ',
;  689          1677  4                                   FI2$S_FILENAME+FI2$S_FILENAMEXT, ATT_BUFFER);
;  690          1678  4                          IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  691          1679  4                          GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  692          1680  4                          THEN CH$MOVE (FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT],
;  693          1681  4                                        ATT_BUFFER + FI2$S_FILENAME);
;  694          1682  4                          MAKE_NAMEBLOCK (FI2$S_FILENAME, ATT_BUFFER, NAME_BLOCK);
;  695          1683  4                          ADDRESS = .T;
;  696          1684  3                          END;
;  697          1685  3
;  698          1686  4          [ACT_R50_NAME]: BEGIN
;  699          1687  4                          T = NAME_BLOCK[NMB$W_NAME];
;  700          1688  4                          CH$COPY (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ' ',
;  701          1689  4                                   FI2$S_FILENAME+FI2$S_FILENAMEXT, ATT_BUFFER);
;  702          1690  4                          IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  703          1691  4                          GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  704          1692  4                          THEN
;  705          1693  5                              BEGIN
;  706          1694  5                              CH$MOVE (FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT],
;  707          1695  5                                       ATT_BUFFER + FI2$S_FILENAME);
;  708          1696  5                              MAKE_NAMEBLOCK (FI2$S_FILENAME+FI2$S_FILENAMEXT,
;  709          1697  5                                              ATT_BUFFER, NAME_BLOCK);
;  710          1698  5                              END
;  711          1699  4                          ELSE MAKE_NAMEBLOCK (FI2$S_FILENAME, ATT_BUFFER, NAME_BLOCK);
;  712          1700  4                          ADDRESS = .T;
;  713          1701  3                          END;
;  714          1702  3
;  715          1703  4          [ACT_R50_TYPE]: BEGIN
;  716          1704  4                          T = NAME_BLOCK[NMB$W_TYPE];
;  717          1705  4                          CH$COPY (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ' ',
```

```
;  718          1706  4                          FI2$S_FILENAME+FI2$S_FILENAMEXT, ATT_BUFFER);
;  719          1707  4                  IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  720          1708  4                  GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  721          1709  4                  THEN
;  722          1710  5                      BEGIN
;  723          1711  5                      CH$MOVE (FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT],
;  724          1712  5                               ATT_BUFFER + FI2$S_FILENAME);
;  725          1713  5                      MAKE_NAMEBLOCK (FI2$S_FILENAME+FI2$S_FILENAMEXT,
;  726          1714  5                                      ATT_BUFFER, NAME_BLOCK);
;  727          1715  5                      END
;  728          1716  4                  ELSE MAKE_NAMEBLOCK (FI2$S_FILENAME, ATT_BUFFER, NAME_BLOCK);
;  729          1717  4                  ADDRESS = .T;
;  730          1718  3                  END;
;  731          1719  3
;  732          1720  4          [ACT_R50_VER]:  BEGIN
;  733          1721  4                  T = NAME_BLOCK[NMB$W_VERSION];
;  734          1722  4                  CH$COPY (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ' ',
;  735          1723  4                           FI2$S_FILENAME+FI2$S_FILENAMEXT, ATT_BUFFER);
;  736          1724  4                  IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  737          1725  4                  GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  738          1726  4                  THEN
;  739          1727  5                      BEGIN
;  740          1728  5                      CH$MOVE (FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT],
;  741          1729  5                               ATT_BUFFER + FI2$S_FILENAME);
;  742          1730  5                      MAKE_NAMEBLOCK (FI2$S_FILENAME+FI2$S_FILENAMEXT,
;  743          1731  5                                      ATT_BUFFER, NAME_BLOCK);
;  744          1732  5                      END
;  745          1733  4                  ELSE MAKE_NAMEBLOCK (FI2$S_FILENAME, ATT_BUFFER, NAME_BLOCK);
;  746          1734  4                  ADDRESS = .T;
;  747          1735  3                  END;
;  748          1736  3
;  749          1737  4          [ACT_DATE]:     BEGIN
;  750          1738  4                  GET_TIME (ATT_BUFFER, .ADDRESS);
;  751          1739  4                  ADDRESS = ATT_BUFFER;
;  752          1740  3                  END;
;  753          1741  3
;  754          1742  4          [ACT_DATES]:    BEGIN
;  755          1743  4                  ATT_BUFFER<0,16> = .ADDRESS[FI2$W_REVISION];
;  756          1744  4                  GET_TIME (ATT_BUFFER+02, ADDRESS[FI2$Q_REVDATE]);
;  757          1745  4                  GET_TIME (ATT_BUFFER+15, ADDRESS[FI2$Q_CREDATE]);
;  758          1746  4                  GET_TIME (ATT_BUFFER+28, ADDRESS[FI2$Q_EXPDATE]);
;  759          1747  4                  ADDRESS = ATT_BUFFER;
;  760          1748  3                  END;
;  761          1749  3
;  762          1750  4          [ACT_ACL]:      BEGIN
;  763          1751  4                  IF .CODE + 1 EQL ATR$C_ADDACLENT
;  764          1752  4                  OR .CODE + 1 EQL ATR$C_DELACLENT
;  765          1753  4                  OR .CODE + 1 EQL ATR$C_MODACLENT
;  766          1754  4                  OR .CODE + 1 EQL ATR$C_DELETEACL
;  767          1755  4                  THEN STATUS = 1
;  768          1756  4                  ELSE STATUS = ACL_DISPATCH (.CODE, .ADDRESS, .COUNT, .P);
;  769          1757  4                  IF NOT .STATUS
;  770          1758  4                  THEN (CURRENT_FIB[FIB$L_ACL_STATUS] = .STATUS; STATUS = 1);
;  771          1759  4                  COUNT = 0;
;  772          1760  3                  END;
;  773          1761  3
;  774          1762  4          [ACT_RESERVED]: BEGIN
```

```
;  775          1763  4                          CH$FILL (0, .COUNT, .P);
;  776          1764  5                          IF ((.P)<0,16> = $BYTEOFFSET (FH2$W_CHECKSUM) -
;  777          1765  4                                              .LOCAL_HEADER[FH2$B_RSOFFSET] * 2) GTR 0
;  778          1766  4                          THEN CH$COPY (.(.P)<0,16>, .ADDRESS, 0, .COUNT, .P + 2);
;  779          1767  4                          COUNT = 0;
;  780          1768  3                          END;
;  781          1769  3
;  782          1770  4              [ACT_ACMODE]:   BEGIN
;  783          1771  4                          (.P-1)<0,8> = .IO_PACKET[IRP$V_MODE];
;  784          1772  4                          ACCESS_MODE = MAXU (.IO_PACKET[IRP$V_MODE], .(.P)<0,8>);
;  785          1773  4                          COUNT = 0;
;  786          1774  3                          END;
;  787          1775  3
;  788          1776  3              TES;
;  789          1777  3              CH$MOVE (.COUNT, .ADDRESS, .P); ! finally copy the attribute
;  790          1778  3
;  791          1779  3
;  792          1780  3
;  793          1781  2          END;                                ! end of loop
;  794          1782  2
;  795          1783  2 ! Stop metering of this subfunction
;  796          1784  2 !
;  797          1785  2
;  798          1786  2 PMS_END_SUB ();
;  799          1787  2
;  800          1788  2 RETURN .STATUS;
;  801          1789  2
;  802          1790  1 END;                                         ! end of routine READ_ATTRIB


                                                .TITLE  RWATTR
                                                .IDENT  \V04-000\

                                                .PSECT  $CODE$,NOWRT,2

                09  3C  02  02  00000 P.AAA:    .BYTE   2, 2, 60, 9
                    0005  0006  00004           .WORD   6, 5
                0A  40  02  02  00008           .BYTE   2, 2, 64, 10
                    0003  0002  0000C           .WORD   2, 3
                02  34  02  04  00010           .BYTE   4, 2, 52, 2
                    0004  0004  00014           .WORD   4, 4
                02  14  02  04  00018           .BYTE   4, 2, 20, 2
                    0020  0020  0001C           .WORD   32, 32
                06  00  03  00  00020           .BYTE   0, 3, 0, 6
                    000A  0014  00024           .WORD   20, 10
                07  00  03  00  00028           .BYTE   0, 3, 0, 7
                    0004  0014  0002C           .WORD   20, 4
                08  00  03  00  00030           .BYTE   0, 3, 0, 8
                    0002  0014  00034           .WORD   20, 2
                0B  26  03  02  00038           .BYTE   2, 3, 38, 11
                    0007  0008  0003C           .WORD   8, 7
                03  00  01  01  00040           .BYTE   1, 1, 0, 3
                    0020  0000  00044           .WORD   0, 32
                02  00  02  01  00048           .BYTE   1, 2, 0, 2
                    0200  0000  0004C           .WORD   0, 512
                05  00  00  00  00050           .BYTE   0, 0, 0, 5
                    0002  0000  00054           .WORD   0, 2
```

```
04  00  00  00  00058          .BYTE   0, 0, 0, 4
    0050  0050  0005C          .WORD   80, 80
0C  00  03  02  00060          .BYTE   2, 3, 0, 12
    0023  002E  00064          .WORD   46, 35
00  00  00  00  00068          .BYTE   0, 0, 0, 0
    000E  0000  0006C          .WORD   0, 14
00  00  00  00  00070          .BYTE   0, 0, 0, 0
    0001  0000  00074          .WORD   0, 1
14  00  03  00  00078          .BYTE   0, 3, 0, 20
    0056  0014  0007C          .WORD   20, 86
02  16  03  02  00080          .BYTE   2, 3, 22, 2
    0008  0008  00084          .WORD   8, 8
02  1E  03  02  00088          .BYTE   2, 3, 30, 2
    0008  0008  0008C          .WORD   8, 8
02  26  03  02  00090          .BYTE   2, 3, 38, 2
    0008  0008  00094          .WORD   8, 8
02  2E  03  02  00098          .BYTE   2, 3, 46, 2
    0008  0008  0009C          .WORD   8, 8
0D  3C  02  02  000A0          .BYTE   2, 2, 60, 13
    0004  0004  000A4          .WORD   4, 4
02  40  02  02  000A8          .BYTE   2, 2, 64, 2
    0002  0002  000AC          .WORD   2, 2
02  38  02  02  000B0          .BYTE   2, 2, 56, 2
    0002  0002  000B4          .WORD   2, 2
13  3B  02  02  000B8          .BYTE   2, 2, 59, 19
    0001  0001  000BC          .WORD   1, 1
04  00  02  02  000C0          .BYTE   2, 2, 0, 4
    0008  0008  000C4          .WORD   8, 8
02  3C  02  01  000C8          .BYTE   1, 2, 60, 2
    0004  0004  000CC          .WORD   4, 4
02  42  01  01  000D0          .BYTE   1, 1, 66, 2
    0002  0000  000D4          .WORD   0, 2
02  42  02  02  000D8          .BYTE   2, 2, 66, 2
    0006  0006  000DC          .WORD   6, 6
02  48  02  02  000E0          .BYTE   2, 2, 72, 2
    0002  0002  000E4          .WORD   2, 2
0E  00  00  02  000E8          .BYTE   2, 0, 0, 14
    0001  0000  000EC          .WORD   0, 1
0F  00  05  04  000F0          .BYTE   4, 5, 0, 15
    0200  0000  000F4          .WORD   0, 512
0F  00  05  04  000F8          .BYTE   4, 5, 0, 15
    00FF  0000  000FC          .WORD   0, 255
0F  00  05  04  00100          .BYTE   4, 5, 0, 15
    00FF  0000  00104          .WORD   0, 255
0F  00  05  01  00108          .BYTE   1, 5, 0, 15
    00FF  0000  0010C          .WORD   0, 255
0F  00  05  01  00110          .BYTE   1, 5, 0, 15
    00FF  0000  00114          .WORD   0, 255
0F  00  05  04  00118          .BYTE   4, 5, 0, 15
    00FF  0000  0011C          .WORD   0, 255
0F  00  05  01  00120          .BYTE   1, 5, 0, 15
    0200  0000  00124          .WORD   0, 512
0F  00  05  01  00128          .BYTE   1, 5, 0, 15
    0004  0000  0012C          .WORD   0, 4
0F  00  05  01  00130          .BYTE   1, 5, 0, 15
    00FF  0000  00134          .WORD   0, 255
10  00  06  02  00138          .BYTE   2, 6, 0, 16
```

```
                017C  0000  0013C           .WORD   0, 380
             02 4C  02  01  00140           .BYTE   1, 2, 76, 2                                                     ;
                0004  0004  00144           .WORD   4, 4
             01 00  00  01  00148           .BYTE   1, 0, 0, 1
                0000  0000  0014C           .WORD   0, 0
             02 00  07  01  00150           .BYTE   1, 7, 0, 2
                0004  0000  00154           .WORD   0, 4
             02 01  07  01  00158           .BYTE   1, 7, 1, 2
                00FF  0000  0015C           .WORD   0, 255
             12 00  00  01  00160           .BYTE   1, 0, 0, 18
                0001  0000  00164           .WORD   0, 1
             02 00  08  01  00168           .BYTE   1, 8, 0, 2
                0200  0000  0016C           .WORD   0, 512
             11 58  02  06  00170           .BYTE   6, 2, 88, 17
                0014  0014  00174           .WORD   20, 20
             C^ 00  00  00  00178           .BYTE   0, 0, 0, 0
                0000  0000  0017C           .WORD   0, 0
                  00000200  00180 P.AAB:    .LONG   512
                        20  00184 P.AAC:    .ASCII  \ \                                                             ;

                                    ATC=                P.AAA
                                            .EXTRN  PMS_START_SUB, PMS_END_SUB
                                            .EXTRN  GET_TIME, MAKE_NAMEBLOCK
                                            .EXTRN  ACL_DISPATCH

                            0BFC 00000      .ENTRY  READ_ATTRIB, Save R2,R3,R4,R5,R6,R7,R8,R9,-                     ; 1421
                                                    R11
                    5E      FF60    CE  9E 00002    MOVAB   -160(SP), SP                                            ;
                1C  AE        80    AA  9E 00007    MOVAB   -128(BASE), 28(SP)                                      ; 1466
                18  AE        90    AA  9E 0000C    MOVAB   -112(BASE), 24(SP)
                14  AE      02E8    CA  9E 00011    MOVAB   744(BASE), 20(SP)
                    6D      03A6    CF  DE 00017    MOVAL   66$, (FP)
                                    09  DD 0001C    PUSHL   #9                                                      ; 1483
            0000G   CF              01  FB 0001E    CALLS   #1, PMS_START_SUB                                       ;
                10  AE              01  D0 00023    MOVL    #1, STATUS                                              ; 1485
                    57          04  AC  D0 00027    MOVL    HEADER, LOCAL_HEADER                                    ; 1486
                    50          18  BE  D0 0002B    MOVL    @24(SP), R0                                             ; 1492
                2A  A0              02  88 0002F    BISB2   #2, 42(R0)
                    50          18  BE  D0 00033    MOVL    @24(SP), R0                                             ; 1493
OC  AE      0B  A0                  02  00  EF 00037    EXTZV   #0, #2, 11(R0), ACCESS_MODE
                    50          18  BE  D0 0003E    MOVL    @24(SP), R0                                             ; 1500
                08  AE          32  A0  3C 00042    MOVZWL  50(R0), 8(SP)
                04  AE              04  D0 00047    MOVL    #4, I
                                035E  31  00004B    BRW     63$                                                     ;
                    51          04  AE  D0 0004E 1$:    MOVL    I, R1                                               ; 1517
                    50          08 BC41 7E 00052    MOVAQ   @ABD[R1], R0                                            ;
                    59              60  3C 00057    MOVZWL  (R0), P                                                 ;
                    59              50  C0 0005A    ADDL2   R0, P                                                   ;
                    6E          02  A0  3C 0005D    MOVZWL  2(R0), COUNT                                            ; 1518
                    56              69  9A 00061    MOVZBL  (P), CODE                                               ; 1519
                                    56  D7 00064    DECL    CODE                                                    ;
                    89          0C  AE  90 00066    MOVB    ACCESS_MODE, (P)+                                       ; 1520
                    2F              56  D1 0006A    CMPL    CODE, #47                                               ; 1528
                                    13  15 0006D    BLEQ    4$                                                      ;
                    50          18  BE  D0 0006F 2$:    MOVL    @24(SP), R0                                         ; 1531
                32  A0          04  AE  B0 00073    MOVW    I, 50(R0)                                               ;
                    03          1C  BE  E8 00078    BLBS    @28(SP), 3$                                             ; 1532
```

```
                                   0142  31 0007C          BRW     28$                            ;
                                   013B  31 0007F 3$:      BRW     27$                            ;
                              FDFA CF46  7F 00082 4$:      PUSHAQ  ATC+6[CODE]                    ; 1535
                          50        9E  3C 00087           MOVZWL  @(SP)+, MAX_COUNT              ;
                          50        6E  D1 0008A           CMPL    COUNT, MAX_COUNT               ; 1536
                                    E0  14 0008D           BGTR    2$                             ;
                              FDEA CF46  7F 0008F           PUSHAQ  ATC+3[CODE]                    ; 1547
                          5B        9E  90 00094           MOVB    @(SP)+, ACTION                 ;
                              FDE0 CF46  7F 00097           PUSHAQ  ATC+1[CODE]                    ; 1554
              08          00        9E  8F 0009C           CASEB   @(SP)+, #0, #8                 ;
      0044    001E        0018    0012     000A0 5$:       .WORD   6$-5$,-                        ;
      008A    0080        007A    0074     000A8                   7$-5$,-                        ;
                                  00C5     000B0                   8$-5$,-                        ;
                                                                   10$-5$,-                       ;
                                                                   12$-5$,-                       ;
                                                                   13$-5$,-                       ;
                                                                   14$-5$,-                       ;
                                                                   17$-5$,-                       ;
                                                                   23$-5$                         ;
                          50    20  AE  9E 000B2 6$:       MOVAB   ATT_BUFFER, R0                 ;
                                    70  11 000B6           BRB     16$                            ;
                          50    08  AA  D0 000B8 7$:       MOVL    8(BASE), R0                    ; 1557
                                    6A  11 000BC           BRB     16$                            ;
                              FDBA CF46  7F 000BE 8$:      PUSHAQ  ATC+2[CODE]                    ; 1560
                          51        9E  9A 000C3           MOVZBL  @(SP)+, R1                     ;
                              FDB4 CF46  7F 000C6           PUSHAQ  ATC+4[CODE]                    ;
                          50        9E  3C 000CB           MOVZWL  @(SP)+, R0                     ;
                          51        50  C0 000CE           ADDL2   R0, R1                         ;
                          50        67  9A 000D1           MOVZBL  (LOCAL_HEADER), R0             ; 1561
                          50        02  C4 000D4           MULL2   #2, R0                         ;
                          50        51  D1 000D7           CMPL    R1, R0                         ;
                                    03  1B 000DA           BLEQU   9$                             ;
                          5B        04  90 000DC           MOVB    #4, ACTION                     ; 1562
                          50        57  D0 000DF 9$:       MOVL    LOCAL_HEADER, R0               ; 1563
                                    7F  11 000E2           BRB     22$                            ;
                              FD94 CF46  7F 000E4 10$:     PUSHAQ  ATC+2[CODE]                    ; 1567
                          52        9E  9A 000E9           MOVZBL  @(SP)+, R2                     ;
                              FD8E CF46  7F 000EC           PUSHAQ  ATC+4[CODE]                    ;
                          50        9E  3C 000F1           MOVZWL  @(SP)+, R0                     ;
                          52        50  C0 000F4           ADDL2   R0, R2                         ;
                          51    01  A7  9A 000F7           MOVZBL  1(LOCAL_HEADER), R1            ; 1568
                          51        02  C4 000FB           MULL2   #2, R1                         ;
                          50        67  9A 000FE           MOVZBL  (LOCAL_HEADER), R0             ; 1569
                          50        02  C4 00101           MULL2   #2, R0                         ;
                          51        50  C2 00104           SUBL2   R0, R1                         ;
                          51        52  D1 00107           CMPL    R2, R1                         ;
                                    03  1B 0010A           BLEQU   11$                            ;
                          5B        04  90 0010C           MOVB    #4, ACTION                     ; 1570
                          50        67  9A 0010F 11$:      MOVZBL  (LOCAL_HEADER), R0             ; 1571
                                    10  11 00112           BRB     15$                            ;
                          50    01  A7  9A 00114 12$:      MOVZBL  1(LOCAL_HEADER), R0            ; 1573
                                    0A  11 00118           BRB     15$                            ;
                          50    02  A7  9A 0011A 13$:      MOVZBL  2(LOCAL_HEADER), R0            ; 1574
                                    04  11 0011E           BRB     15$                            ;
                          50    03  A7  9A 00120 14$:      MOVZBL  3(LOCAL_HEADER), R0            ; 1575
                          50      6740  3E 00124 15$:      MOVAW   (LOCAL_HEADER)[R0], R0         ;
                                    4B  11 00128 16$:      BRB     24$                            ;
```

```
                                   FD4E CF46  7F 0012A 17$:   PUSHAQ  ATC+2[CODE]                        ; 1576
          01              00              9E  8F 0012F        CASEB   @(SP)+, #0, #1                     ;
                        000A         0004        00133 18$:   .WORD   19$-18$,-                          ;
                                                                      20$-18$                            ;
                          50      C4      AA  9E 00137 19$:   MOVAB   -60(BASE), R0                      ;
                                          1B  11 0013B        BRB     21$                                ;
                          51      14      BE  9A 0013D 20$:   MOVZBL  @20(SP), R1                        ; 1580
          50 000000FF     8F              51  C3 00141        SUBL3   R1, #255, R0                       ;
                          58      14      AE  D0 00149        MOVL    20(SP), R8                         ; 1581
  50      00              6E              00  2C 0014D        MOVC5   #0, (SP), #0, R0, (R1)[R8]         ;
                                   6148          00152                                                   ;
                          50      14      AE  D0 00154        MOVL    20(SP), R0                         ; 1579
                                   FD20 CF46  7F 00158 21$:   PUSHAQ  ATC+2[CODE]                        ; 1584
                          51              9E  9A 0015D        MOVZBL  @(SP)+, R1                         ;
                          50              51  C2 00160        SUBL2   R1, R0                             ;
                                          10  11 00163 22$:   BRB     24$                                ;
                                          57  DD 00165 23$:   PUSHL   LOCAL_HEADER                       ; 1586
                   0000V  CF              01  FB 00167        CALLS   #1, FID_TO_SPEC                    ;
                          57      04      AA  D0 0016C        MOVL    4(BASE), LOCAL_HEADER              ; 1587
                          50    04E8      CA  9E 00170        MOVAB   1256(BASE), R0                     ; 1585
                                   FD03 CF46  7F 00175 24$:   PUSHAQ  ATC+2[CODE]                        ; 1592
                          58              9E  9A 0017A        MOVZBL  @(SP)+, ADDRESS                    ;
                          58              50  C0 0017D        ADDL2   R0, ADDRESS                        ;
                14        00              5B  8F 00180        CASEB   ACTION, #0, #20                    ; 1598
  005B        0224      002C         0222        00184 25$:   .WORD   61$-25$,-                          ;
  0144        0114      00A5         00AC        0018C                26$-25$,-                          ;
  0162        00E2      00BD         014A        00194                62$-25$,-                          ;
  0196        00B5      0224         0169        0019C                31$-25$,-                          ;
  0224        01FE      0043         01D9        001A4                33$-25$,-                          ;
                                     00EE        001AC                32$-25$,-                          ;
                                                                      43$-25$,-                          ;
                                                                      45$-25$,-                          ;
                                                                      46$-25$,-                          ;
                                                                      37$-25$,-                          ;
                                                                      40$-25$,-                          ;
                                                                      49$-25$,-                          ;
                                                                      50$-25$,-                          ;
                                                                      62$-25$,-                          ;
                                                                      35$-25$,-                          ;
                                                                      54$-25$,-                          ;
                                                                      58$-25$,-                          ;
                                                                      29$-25$,-                          ;
                                                                      59$-25$,-                          ;
                                                                      62$-25$,-                          ;
                                                                      42$-25$                            ;
                                          2D  11 001AE        BRB     30$                                ; 1601
                          50      18      BE  D0 001B0 26$:   MOVL    @24(SP), R0                        ; 1604
                  32      A0      04      AE  B0 001B4        MOVW    I, 50(R0)                          ;
                          04      1C      BE  E9 001B9        BLBC    @28(SP), 28$                       ; 1605
                  1C      BE              34  B0 001BD 27$:   MOVW    #52, @28(SP)                       ;
                                  10      AE  D4 001C1 28$:   CLRL    STATUS                             ;
                                        01F0  31 001C4        BRW     65$                                ; 1603
  6E      00              6E              00  2C 001C7 29$:   MOVC5   #0, (SP), #0, COUNT, (P)           ; 1613
                                          69     001CC                                                   ;
                          50      04      BC  9A 001CD        MOVZBL  @HEADER, R0                        ; 1614
                          50              02  C4 001D1        MULL2   #2, R0                             ;
               00000050   8F              50  D1 001D4        CMPL    R0, #80                            ;
```

```
                                                61  1A 001DB          BGTRU   36$                                        ;
                                                57  11 001DD 30$:     BRB     34$                                        ; 1615
              20   AE        30   A8            10  9C 001DF 31$:     ROTL    #16, 48(ADDRESS), ATT_BUFFER               ; 1619
              24   AE        38   A8            10  9C 001E5          ROTL    #16, 56(ADDRESS), ATT_BUFFER+4             ; 1620
                             28   AE      1A    A8  90 001EB          MOVB    26(ADDRESS), ATT_BUFFER+8                  ; 1621
                             29   AE      1E    A8  90 001F0          MOVB    30(ADDRESS), ATT_BUFFER+9                  ; 1622
                             2A   AE            58  D0 001F5          MOVL    ADDRESS, ATT_BUFFER+10                     ; 1623
                                          2E    AE  B4 001F9          CLRW    ATT_BUFFER+14                              ; 1624
                             30   AE      1A    A8  B0 001FC          MOVW    26(ADDRESS), ATT_BUFFER+16                 ; 1625
                             32   AE      1E    A8  B0 00201          MOVW    30(ADDRESS), ATT_BUFFER+18                 ; 1626
                             34   AE      1C    A8  B0 00206          MOVW    28(ADDRESS), ATT_BUFFER+20                 ; 1627
                             36   AE      20    A8  B0 0020B          MOVW    32(ADDRESS), ATT_BUFFER+22                 ; 1628
                                          38    AE  7C 00210          CLRQ    ATT_BUFFER+24                              ; 1629
                                  50      0C    AA  D0 00213          MOVL    12(BASE), R0                               ; 1631
                                                56  13 00217          BEQL    41$                                        ;
                             38   AE      24    A0  D0 00219          MOVL    36(R0), ATT_BUFFER+24                      ; 1634
                                  50      0C    AA  D0 0021E          MOVL    12(BASE), R0                               ; 1635
                             3C   AE      28    A0  D0 00222          MOVL    40(R0), ATT_BUFFER+28                      ;
                                                46  11 00227          BRB     41$                                        ; 1637
                                  58    FDCE    CF  9E 00229 32$:     MOVAB   P.AAB, ADDRESS                             ; 1640
                                                0E  11 0022E          BRB     36$                                        ;
        6E         00             6E            00  2C 00230 33$:     MOVC5   #0, (SP), #0, COUNT, (P)                   ; 1643
                                                69     00235                                                             ;
                                              016D  31 00236 34$:     BRW     61$                                        ; 1644
                                  58    FDC2    CF  9E 00239 35$:     MOVAB   P.AAC, ADDRESS                             ; 1648
                                              0167  31 0023E 36$:     BRW     62$                                        ; 1598
                             20   AE            68  90 00241 37$:     MOVB    (ADDRESS), ATT_BUFFER                      ; 1653
                             21   AE      02    A8  90 00245          MOVB    2(ADDRESS), ATT_BUFFER+1                   ; 1654
                                          01    A8  95 0024A          TSTB    1(ADDRESS)                                 ; 1655
                                                05  12 0024D          BNEQ    38$                                        ;
                                          03    A8  95 0024F          TSTB    3(ADDRESS)                                 ; 1656
                                                06  13 00252          BEQL    39$                                        ;
                             20   AE    FFFF    8F  B0 00254 38$:     MOVW    #65535, ATT_BUFFER                         ; 1659
                             22   AE      40    A7  B0 0025A 39$:     MOVW    64(LOCAL_HEADER), ATT_BUFFER+2             ; 1662
                             24   AE      34    A7  90 0025F          MOVB    52(LOCAL_HEADER), ATT_BUFFER+4             ; 1663
                                                09  11 00264          BRB     41$                                        ; 1664
                             20   AE            68  B0 00266 40$:     MOVW    (ADDRESS), ATT_BUFFER                      ; 1669
                             22   AE      34    A7  90 0026A          MOVB    52(LOCAL_HEADER), ATT_BUFFER+2             ; 1670
                                              00A1  31 0026F 41$:     BRW     52$                                        ; 1671
                                  5B      20    AE  9E 00272 42$:     MOVAB   ATT_BUFFER, T                              ; 1675
 0056   8F         20             68            14  2C 00276          MOVC5   #20, (ADDRESS), #32, #86, ATT_BUFFER       ; 1676
                                          20    AE     0027D                                                             ;
                                  50      01    A7  9A 0027F          MOVZBL  1(LOCAL_HEADER), R0                        ; 1678
                                  51            67  9A 00283          MOVZBL  (LOCAL_HEADER), R1                         ;
                                  50            51  C2 00286          SUBL2   R1, R0                                     ;
                                  3C            50  D1 00289          CMPL    R0, #60                                    ; 1679
                                                46  1F 0028C          BLSSU   47$                                        ;
              34   AE        36   A8    0042    8F  28 0028E          MOVC3   #66, 54(ADDRESS), ATT_BUFFER+20            ; 1681
                                                3C  11 00296          BRB     47$                                        ; 1682
                                  5B      7E    AE  9E 00298 43$:     MOVAB   NAME_BLOCK+6, T                            ; 1687
 0056   8F         20             68            14  2C 0029C 44$:     MOVC5   #20, (ADDRESS), #32, #86, ATT_BUFFER       ; 1688
                                          20    AE     002A3                                                             ;
                                  50      01    A7  9A 002A5          MOVZBL  1(LOCAL_HEADER), R0                        ; 1690
                                  51            67  9A 002A9          MOVZBL  (LOCAL_HEADER), R1                         ;
                                  50            51  C2 002AC          SUBL2   R1, R0                                     ;
                                  3C            50  D1 002AF          CMPL    R0, #60                                    ; 1691
                                                20  1F 002B2          BLSSU   47$                                        ;
```

```
                34   AE        36   A8      0042   8F   28 002B4          MOVC3   #66, 54(ADDRESS), ATT_BUFFER+20          ; 1695
                                            78   AE   9F 002BC          PUSHAB  NAME_BLOCK                               ; 1696
                                            24   AE   9F 002BF          PUSHAB  ATT_BUFFER
                                       7E        56   8F   9A 002C2          MOVZBL  #86, -(SP)
                                                 14   11 002C6          BRB     48$                                      ; 1699
                                       5B        E4   AD   9E 002C8 45$:     MOVAB   NAME_BLOCK+12, T                         ; 1704
                                                 CE   11 002CC          BRB     44$                                      ; 1705
                                       5B        E6   AD   9E 002CE 46$:     MOVAB   NAME_BLOCK+14, T                         ; 1721
                                                 C8   11 002D2          BRB     44$                                      ; 1722
                                                 78   AE   9F 002D4 47$:     PUSHAB  NAME_BLOCK                               ; 1733
                                                 24   AE   9F 002D7          PUSHAB  ATT_BUFFER
                                                 14   DD 002DA          PUSHL   #20
                                  0000G  CF              03   FB 002DC 48$:     CALLS   #3, MAKE_NAMEBLOCK
                                         58              5B   D0 002E1          MOVL    T, ADDRESS                       ; 1734
                                                 31   11 002E4          BRB     53$                                      ; 1598
                                                 58   DD 002E6 49$:     PUSHL   ADDRESS                                  ; 1738
                                                 24   AE   9F 002E8          PUSHAB  ATT_BUFFER
                                                 21   11 002EB          BRB     51$
                                    20   AE        14   A8   B0 002ED 50$:     MOVW    20(ADDRESS), ATT_BUFFER          ; 1743
                                                 1E   A8   9F 002F2          PUSHAB  30(ADDRESS)                              ; 1744
                                                 26   AE   9F 002F5          PUSHAB  ATT_BUFFER+2
                                  0000G  CF              02   FB 002F8          CALLS   #2, GET_TIME
                                                 16   A8   9F 002FD          PUSHAB  22(ADDRESS)                              ; 1745
                                                 33   AE   9F 00300          PUSHAB  ATT_BUFFER+15
                                  0000G  CF              02   FB 00303          CALLS   #2, GET_TIME
                                                 26   A8   9F 00308          PUSHAB  38(ADDRESS)                              ; 1746
                                                 40   AE   9F 0030B          PUSHAB  ATT_BUFFER+28
                                  0000G  CF              02   FB 0030E 51$:     CALLS   #2, GET_TIME
                                         58        20   AE   9E 00313 52$:     MOVAB   ATT_BUFFER, ADDRESS              ; 1747
                                              008E   31 00317 53$:     BRW     62$                                      ; 1598
                                       50        01   A6   9E 0031A 54$:     MOVAB   1(R6), R0                        ; 1751
                                       1F             50   D1 0031E          CMPL    R0, #31
                                                      0F   13 00321          BEQL    55$
                                       20             50   D1 00323          CMPL    R0, #32                          ; 1752
                                                      0A   13 00326          BEQL    55$
                                       21             50   D1 00328          CMPL    R0, #33                          ; 1753
                                                      05   13 0032B          BEQL    55$
                                       24             50   D1 0032D          CMPL    R0, #36                          ; 1754
                                                      06   12 00330          BNEQ    56$
                                  10   AE             01   D0 00332 55$:     MOVL    #1, STATUS                       ; 1755
                                                      12   11 00336          BRB     57$
                                                      59   DD 00338 56$:     PUSHL   P                                ; 1756
                                            04        AE   DD 0033A          PUSHL   COUNT
                                          0140        8F   BB 0033D          PUSHR   #^M<R6,R8>
                                  0000G  CF              04   FB 00341          CALLS   #4, ACL_DISPATCH
                                    10   AE             50   D0 00346          MOVL    R0, STATUS
                                         58        10   AE   E8 0034A 57$:     BLBS    STATUS, 61$                      ; 1757
                                         50        10   AA   D0 0034E          MOVL    16(BASE), R0                     ; 1758
                                    34   A0        10   AE   D0 00352          MOVL    STATUS, 52(R0)
                                    10   AE             01   D0 00357          MOVL    #1, STATUS
                                                      49   11 0035B          BRB     61$                              ; 1759
6E                  00                  6E             00   2C 0035D 58$:     MOVC5   #0, (SP), #0, COUNT, (P)         ; 1763
                                                      69        00362
                                       50        03   A7   9A 00363          MOVZBL  3(LOCAL_HEADER), R0              ; 1765
                                       50             50   CE 00367          MNEGL   R0, R0                           ; 1764
                                       50             02   C4 0036A          MULL2   #2, R0
                                       50        01FE CO   9E 0036D          MOVAB   510(R0), R0
```

```
                        69              50  B0 00372          MOVW    R0, (P)
                                        50  D5 00375          TSTL    R0                                          ; 1765
                                        2D  15 00377          BLEQ    61$
        6E              00              69  2C 00379          MOVC5   (P), (ADDRESS), #0, COUNT, 2(P)             ; 1766
                                    02  A9    0037E
                                        24  11 00380          BRB     61$                                         ; 1767
                        50          18  BE  D0 00382 59$:     MOVL    @24(SP), R0                                 ; 1771
        51      0B      A0              02  EF 00386          EXTZV   #0, #2, 11(R0), R1
                                FF      A9  51  90 0038C      MOVB    R1, -1(P)
                        50          18  BE  D0 00390          MOVL    @24(SP), R0                                 ; 1772
        50      0B      A0              02  EF 00394          EXTZV   #0, #2, 11(R0), R0
                        50              69  91 0039A          CMPB    (P), R0
                                        03  1B 0039D          BLEQU   60$
                        50              69  9A 0039F          MOVZBL  (P), R0
                                0C      AE  50  D0 003A2 60$:  MOVL    R0, ACCESS_MODE
                                            6E  D4 003A6 61$:  CLRL    COUNT                                       ; 1773
                    69                      6E  28 003A8 62$:  MOVC3   COUNT, (ADDRESS), (P)                       ; 1777
                    02          04  AE  08  AE  F2 003AC 63$:  AOBLSS  8(SP), I, 64$                               ; 1500
                                            03  11 003B2      BRB     65$
                                        FC97  31 003B4 64$:    BRW     1$
                            0000G   CF      00  FB 003B7 65$:  CALLS   #0, PMS_END_SUB                             ; 1786
                        50          10  AE  D0 003BC          MOVL    STATUS, R0                                  ; 1788
                                            04 003C0          RET                                                 ; 1790
                                          0000 003C1 66$:     .WORD   Save nothing                                ; 1466
                                        7E  D4 003C3          CLRL    -(SP)
                                        5E  DD 003C5          PUSHL   SP
                        7E          04  AC  7D 003C7          MOVQ    4(AP), -(SP)
                            0000V   CF      03  FB 003CB      CALLS   #3, READ_HANDLER
                                            04 003D0          RET

; Routine Size:  977 bytes,    Routine Base:  $CODE$ + 0185
```

```
;  804          1791  1 GLOBAL ROUTINE FID_TO_SPEC (HEADER) : L_NORM NOVALUE =
;  805          1792  1
;  806          1793  1 !++
;  807          1794  1 !
;  808          1795  1 ! FUNCTIONAL DESCRIPTION:
;  809          1796  1 !
;  810          1797  1 !       This routine converts the specified file-ID (contained in the header
;  811          1798  1 !       supplied) to a full file specification.  Because the RMS limit of
;  812          1799  1 !       256 bytes is NOT enforced, it is actually possible to get a file
;  813          1800  1 !       spec string of 444 bytes.
;  814          1801  1 !
;  815          1802  1 !       If a file pointed to by the backlink does not exist, translation
;  816          1803  1 !       stops, and a question mark is inserted in place of the directory
;  817          1804  1 !       name.
;  818          1805  1 !
;  819          1806  1 ! CALLING SEQUENCE:
;  820          1807  1 !       FID_TO_SPEC (ARG1)
;  821          1808  1 !
;  822          1809  1 ! INPUT PARAMETERS:
;  823          1810  1 !       ARG1: address of the file header
;  824          1811  1 !
;  825          1812  1 ! IMPLICIT INPUTS:
;  826          1813  1 !       none
;  827          1814  1 !
;  828          1815  1 ! OUTPUT PARAMETERS:
;  829          1816  1 !       none
;  830          1817  1 !
;  831          1818  1 ! IMPLICIT OUTPUTS:
;  832          1819  1 !       none
;  833          1820  1 !
;  834          1821  1 ! ROUTINE VALUE:
;  835          1822  1 !       none
;  836          1823  1 !
;  837          1824  1 ! SIDE EFFECTS:
;  838          1825  1 !       none
;  839          1826  1 !
;  840          1827  1 !--
;  841          1828  1
;  842          1829  2 BEGIN
;  843          1830  2
;  844          1831  2 MAP
;  845          1832  2         HEADER          : REF BBLOCK;           ! Address of the header
;  846          1833  2
;  847          1834  2 LINKAGE
;  848          1835  2         L_CVT_DEVNAM    = JSB (REGISTER = 0,    ! Buffer length
;  849          1836  2                                REGISTER = 1,    ! Buffer address
;  850          1837  2                                REGISTER = 4,    ! Cluster node conversion flag
;  851          1838  2                                REGISTER = 5;    ! UCB address
;  852          1839  2                                REGISTER = 1);   ! Length of converted name
;  853          1840  2
;  854          1841  2 LITERAL
;  855          1842  2         NODE_LEN        = 15,                   ! Maximum cluster node name length
;  856          1843  2         DEVNAME_LEN     = 15,                   ! Maximum device name length
;  857          1844  2         UNIT_LEN        = 5,                    ! Maximum unit number length
;  858          1845  2         FILENAME_LEN    = 39,                   ! Maximum file name length
;  859          1846  2         FILETYPE_LEN    = 39,                   ! Maximum file type length
;  860          1847  2         FILEVER_LEN     = 5,                    ! Maximum version length
```

```
;  861          1848  2
;  862          1849  2 ! The maximum number of directory levels to traverse is calculated from the
;  863          1850  2 ! available storage (for FULL_FILE_SPEC).
;  864          1851  2
;  865          1852  4         MAX_DIR_LEVEL   = (1022 - (NODE_LEN + DEVNAME_LEN + UNIT_LEN +
;  866          1853  2                                   FILENAME_LEN + FILETYPE_LEN + FILEVER_LEN + 1)) /
;  867          1854  2                           (FILENAME_LEN + 1),
;  868          1855  2
;  869          1856  2 ! Determine the maximum sizes of the various portions of a file specification.
;  870          1857  2
;  871          1858  2         FULLDEV_LEN     = NODE_LEN + 1 +        ! Full device spec length
;  872          1859  2                           DEVNAME_LEN + UNIT_LEN + 1,
;  873          1860  2         FULLDIR_LEN     = 1 + (FILENAME_LEN + 1) * (MAX_DIR_LEVEL + 1), ! Max dir spec length
;  874          1861  2         FULLFILE_LEN    = FILENAME_LEN + 1 +    ! Max file name length
;  875          1862  2                           FILETYPE_LEN + 1 +    !   including type & version
;  876          1863  2                           FILEVER_LEN,
;  877          1864  2         FULLSPEC_LEN    = 2 + FULLDEV_LEN +     ! Maximum file spec length
;  878          1865  2                           FULLDIR_LEN +         ! Includes word size prefix
;  879          1866  2                           FULLFILE_LEN;
;  880          1867  2
;  881          1868  2 LOCAL
;  882          1869  2         DEVICE_LEN,                             ! Length of the device name
;  883          1870  2         DIR_HEADER      : REF BBLOCK,           ! Current directory file header
;  884          1871  2         IDENT_AREA      : REF BBLOCK,           ! Address of header ident area
;  885          1872  2         DIR_ID          : BBLOCK [FID$C_LENGTH],        ! Directory FID
;  886          1873  2         END_NAME;                               ! End of the directory/file name
;  887          1874  2
;  888          1875  2 BIND_COMMON;
;  889          1876  2
;  890          1877  2 EXTERNAL ROUTINE
;  891          1878  2         READ_HEADER     : L_NORM,               ! Read & validate file header
;  892          1879  2         SERIAL_FILE     : L_NORM,               ! Synchronization locking
;  893          1880  2         RELEASE_SERIAL_LOCK : L_NORM NOVALUE,   ! Release synchronization lock
;  894          1881  2         IOC$CVT_DEVNAM  : L_CVT_DEVNAM ADDRESSING_MODE (GENERAL);
;  895          1882  2
;  896          1883  2 BIND
;  897          1884  2         DEVICE_NAME     = FULL_FILE_SPEC        : VECTOR [,BYTE],       ! Device name spec
;  898          1885  2         DIR_NAME        = FULL_FILE_SPEC[FULLDEV_LEN + 1]       : VECTOR [,BYTE],       ! Directory name sto
;  899          1886  2         FILE_NAME       = FULL_FILE_SPEC[FULLDEV_LEN + 1 +      ! File name storage
;  900          1887  2                                         FULLDIR_LEN]   : VECTOR [,BYTE];
;  901          1888  2
;  902          1889  2 ! Initialize all of the necessary storage.
;  903          1890  2
;  904          1891  2 CH$FILL (0, FULLSPEC_LEN, FULL_FILE_SPEC);
;  905          1892  2 CH$MOVE (FID$C_LENGTH, HEADER[FH2$W_BACKLINK], DIR_ID);
;  906          1893  2 FILE_SPEC_LEN[0] = 0;
;  907          1894  2
;  908          1895  2 ! Save the file name from the current file header as the real file name.
;  909          1896  2
;  910          1897  2 IDENT_AREA = .HEADER + .HEADER[FH2$B_IDOFFSET]*2;
;  911          1898  2 CH$MOVE (FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME], FILE_NAME[1]);
;  912          1899  2 IF .HEADER[FH2$B_MPOFFSET] - .HEADER[FH2$B_IDOFFSET]
;  913          1900  2 GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  914          1901  2 THEN CH$MOVE (FI2$S_FILENAMEXT, IDENT_AREA[FI2$T_FILENAMEXT],
;  915          1902  2                                 FILE_NAME[FI2$S_FILENAME + 1]);
;  916          1903  2 INCR J FROM 0 TO FI2$S_FILENAME + FI2$S_FILENAMEXT
;  917          1904  2 DO
```

```
;  918          1905  3      BEGIN
;  919          1906  3      IF .FILE_NAME[.J + 1] EQL ' ' THEN EXITLOOP;
;  920          1907  3      FILE_NAME[0] = .FILE_NAME[0] + 1;
;  921          1908  2      END;
;  922          1909  2
;  923          1910  2  ! Loop through all of the directories back to the MFD, saving the directory
;  924          1911  2  ! names.
;  925          1912  2  ! Before doing so, release the current primary lock index.  This must be
;  926          1913  2  ! done because tracking back up through the directories for this file
;  927          1914  2  ! would cause synchronization deadlocks with other processes coming down
;  928          1915  2  ! through the directories on an access to this same file.
;  929          1916  2  !
;  930          1917  2
;  931          1918  2  IF .PRIM_LCKINDX NEQ 0
;  932          1919  2      AND .PRIM_LCKINDX NEQ .DIR_LCKINDX
;  933          1920  2  THEN
;  934          1921  3      BEGIN
;  935          1922  3      IF .PRIMARY_FCB NEQ 0
;  936          1923  3      THEN
;  937          1924  4          BEGIN
;  938          1925  4          CLEANUP_FLAGS [CLF_PFCB_REF_UP] = 1;
;  939          1926  4          PRIMARY_FCB [FCB$W_REFCNT] = .PRIMARY_FCB [FCB$W_REFCNT] + 1;
;  940          1927  3          END;
;  941          1928  3
;  942          1929  3      RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
;  943          1930  2      END;
;  944          1931  2
;  945          1932  2  DECR J FROM MAX_DIR_LEVEL TO 0
;  946          1933  2  DO
;  947          1934  3      BEGIN
;  948          1935  3      LOCAL
;  949          1936  3          TMPINDX;
;  950          1937  3
;  951          1938  3      IF .DIR_ID[FID$W_NUM] EQL 0 AND .DIR_ID[FID$B_NMX] EQL 0
;  952          1939  3      THEN
;  953          1940  4          BEGIN
;  954          1941  4          IF .J NEQ MAX_DIR_LEVEL
;  955          1942  4          THEN
;  956          1943  5              BEGIN
;  957          1944  5              DIR_NAME[.J * (FILENAME_LEN + 1)] = 1;
;  958          1945  5              DIR_NAME[.J * (FILENAME_LEN + 1) + 1] = '?';
;  959          1946  4              END;
;  960          1947  4          EXITLOOP;
;  961          1948  3          END;
;  962          1949  3
;  963          1950  3      IF .DIR_ID[FID$W_NUM] EQL FID$C_MFD AND .DIR_ID[FID$B_NMX] EQL 0
;  964          1951  3      AND .J NEQ MAX_DIR_LEVEL THEN EXITLOOP;
;  965          1952  3
;  966          1953  3  ! Synchronize on the desired file prior to the read_header call.
;  967          1954  3  ! If this is actually the directory we may have already accessed,
;  968          1955  3  ! then clear tmpindx so that we do not attempt to release it at
;  969          1956  3  ! the end of this loop.
;  970          1957  3  !
;  971          1958  3
;  972          1959  3      STSFLGS [STS_HAD_LOCK] = 0;
;  973          1960  3      TMPINDX = SERIAL_FILE (DIR_ID);
;  974          1961  3
```

```
;  975          1962  3      IF .STSFLGS [STS_HAD_LOCK]
;  976          1963  3      THEN
;  977          1964  3          TMPINDX = 0;
;  978          1965  3
;  979          1966  3      DIR_HEADER = READ_HEADER (DIR_ID, 0);
;  980          1967  3
;  981          1968  3      IF .DIR_HEADER EQL 0
;  982          1969  3      THEN
;  983          1970  4          BEGIN
;  984          1971  4          DIR_NAME[.J * (FILENAME_LEN + 1)] = 1;
;  985          1972  4          DIR_NAME[.J * (FILENAME_LEN + 1) + 1] = '?';
;  986          1973  4          EXITLOOP;
;  987          1974  3          END;
;  988          1975  3      IDENT_AREA = .DIR_HEADER + .DIR_HEADER[FH2$B_IDOFFSET]*2;
;  989          1976  3      CH$MOVE (FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
;  990          1977  3                               DIR_NAME[.J * (FILENAME_LEN + 1) + 1]);
;  991          1978  3      IF .HEADER[FH2$B_MPOFFSET] - .HEADER[FH2$B_IDOFFSET]
;  992          1979  3      GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  993          1980  3      THEN CH$MOVE (FILENAME_LEN - FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAMEXT],
;  994          1981  3                                                   DIR_NAME[.J * (FILENAME_LEN + 1) +
;  995          1982  3                                                            FI2$S_FILENAME + 1]);
;  996          1983  3      INCR K FROM 1 TO FILENAME_LEN
;  997          1984  3      DO
;  998          1985  4          BEGIN
;  999          1986  4          IF .DIR_NAME[.J * (FILENAME_LEN + 1) + .K] EQL '.' THEN EXITLOOP;
; 1000          1987  4          DIR_NAME[.J * (FILENAME_LEN + 1)] = .DIR_NAME[.J * (FILENAME_LEN + 1)] + 1;
; 1001          1988  3          END;
; 1002          1989  3
; 1003          1990  3  ! If all the way back up to the mfd, stop.  Release the temporary lock
; 1004          1991  3  ! on the directory just looked at before exiting the loop.
; 1005          1992  3  !
; 1006          1993  3
; 1007          1994  3      IF .DIR_ID[FID$W_NUM] EQL FID$C_MFD AND .DIR_ID[FID$B_NMX] EQL 0
; 1008          1995  3      THEN
; 1009          1996  4          BEGIN
; 1010          1997  4          IF .TMPINDX NEQ 0
; 1011          1998  4          THEN
; 1012          1999  4              RELEASE_SERIAL_LOCK (.TMPINDX);
; 1013          2000  4
; 1014          2001  4          EXITLOOP;
; 1015          2002  3          END;
; 1016          2003  3
; 1017          2004  3  ! Save the backlink from this one to go around the loop again.
; 1018          2005  3  ! Once the temporary serialization lock is released, the buffer
; 1019          2006  3  ! may be recycled to another process.
; 1020          2007  3  !
; 1021          2008  3
; 1022          2009  3      CH$MOVE (FID$C_LENGTH, DIR_HEADER[FH2$W_BACKLINK], DIR_ID);
; 1023          2010  3
; 1024          2011  3      IF .TMPINDX NEQ 0
; 1025          2012  3      THEN
; 1026          2013  3          RELEASE_SERIAL_LOCK (.TMPINDX);
; 1027          2014  3
; 1028          2015  2      END;
; 1029          2016  2
; 1030          2017  2  ! Get the device name.
; 1031          2018  2
```

```
;  1032          2019  2 IOC$CVT_DEVNAM (FULLDEV_LEN, DEVICE_NAME, 0, .IO_PACKET[IRP$L_UCB], DEVICE_LEN);
;  1033          2020  2 FILE_SPEC_LEN[0] = .FILE_SPEC_LEN[0] + .DEVICE_LEN;
;  1034          2021  2
;  1035          2022  2 ! Start building the directory specification.
;  1036          2023  2
;  1037          2024  2 FULL_FILE_SPEC[.FILE_SPEC_LEN[0]] = '[';
;  1038          2025  2 FILE_SPEC_LEN[0] = .FILE_SPEC_LEN[0] + 1;
;  1039          2026  2
;  1040          2027  2 INCR J FROM 0 TO MAX_DIR_LEVEL
;  1041          2028  2 DO
;  1042          2029  3     BEGIN
;  1043          2030  3     IF (END_NAME = .DIR_NAME[.J * (FILENAME_LEN + 1)]) NEQ 0
;  1044          2031  3     THEN
;  1045          2032  4         BEGIN
;  1046          2033  4         CH$COPY (.END_NAME, DIR_NAME[.J * (FILENAME_LEN + 1) + 1],
;  1047          2034  4                  ' ',
;  1048          2035  4                  .END_NAME + 1, FULL_FILE_SPEC[.FILE_SPEC_LEN[0]]);
;  1049          2036  4         FILE_SPEC_LEN[0] = .FILE_SPEC_LEN[0] + .END_NAME + 1;
;  1050          2037  3         END;
;  1051          2038  2     END;
;  1052          2039  2
;  1053          2040  2 ! Tie off the directory specification.
;  1054          2041  2
;  1055          2042  2 IF .FULL_FILE_SPEC[.FILE_SPEC_LEN[0] - 1] EQL '['
;  1056          2043  2 THEN FILE_SPEC_LEN[0] = .FILE_SPEC_LEN[0] + 1;
;  1057          2044  2
;  1058          2045  2 FULL_FILE_SPEC[.FILE_SPEC_LEN[0] - 1] = ']';
;  1059          2046  2
;  1060          2047  2 ! Now add in the file name.
;  1061          2048  2
;  1062          2049  2 END_NAME = .FILE_NAME[0];
;  1063          2050  2 CH$MOVE (.END_NAME, FILE_NAME[1], FULL_FILE_SPEC[.FILE_SPEC_LEN[0]]);
;  1064          2051  2 FILE_SPEC_LEN[0] = .FILE_SPEC_LEN[0] + .END_NAME;
;  1065          2052  2
;  1066          2053  2 ! Reacquire the primary header serialization lock.
;  1067          2054  2 ! Turn everybody back to the primary file header.
;  1068          2055  2
;  1069          2056  2 IF .PRIM_LCKINDX NEQ 0
;  1070          2057  2 THEN
;  1071          2058  3     BEGIN
;  1072          2059  3     PRIM_LCKINDX = SERIAL_FILE (IF .PRIMARY_FCB NEQ 0
;  1073          2060  3                                 THEN PRIMARY_FCB [FCB$W_FID]
;  1074          2061  3                                 ELSE CURRENT_FIB [FIB$W_FID]);
;  1075          2062  3
;  1076          2063  3     IF TESTBITSC (CLEANUP_FLAGS [CLF_PFCB_REF_UP])
;  1077          2064  3     THEN
;  1078          2065  3         PRIMARY_FCB [FCB$W_REFCNT] = .PRIMARY_FCB [FCB$W_REFCNT] - 1;
;  1079          2066  3
;  1080          2067  3     READ_HEADER (CURRENT_FIB[FIB$W_FID], .PRIMARY_FCB);
;  1081          2068  2     END;
;  1082          2069  2
;  1083          2070  2 RETURN;
;  1084          2071  2
;  1085          2072  1 END;                                    ! End of routine FID_TO_SPEC
```

```
                                                        .EXTRN  READ_HEADER, SERIAL_FILE
                                                        .EXTRN  RELEASE_SERIAL_LOCK
                                                        .EXTRN  IOC$CVT_DEVNAM

                                        0BFC 00000      .ENTRY  FID_TO_SPEC, Save R2,R3,R4,R5,R6,R7,R8,R9,-     ; 1791
                                                                R11                                             ;
                                    5E          10  C2 00002      SUBL2   #16, SP                                 ;
                                            08  AA  9F 00005      PUSHAB  8(BASE)                                 ; 1873
                                        04E8    CA  9F 00008      PUSHAB  1256(BASE)                              ;
                                    59  04EA    CA  9E 0000C      MOVAB   1258(BASE), R9                          ;
                                    58    26    A9  9E 00011      MOVAB   38(R9), R8                              ; 1885
                                    5B  03BF    C9  9E 00015      MOVAB   959(R9), R11                            ; 1887
    0415    8F          00          6E          00  2C 0001A      MOVC5   #0, (SP), #0, #1045, (R9)               ; 1891
                                                69      00021                                                   ;
                                    50      04  AC  D0 00022      MOVL    HEADER, R0                              ; 1892
                10  AE      42     A0          06  28 00026      MOVC3   #6, 66(R0), DIR_ID                      ;
                                            00  BE  B4 0002C      CLRW    @0(SP)                                  ; 1893
                                    50      04  BC  9A 0002F      MOVZBL  @HEADER, R0                             ; 1897
                                    57      04  BC40 3E 00033      MOVAW   @HEADER[R0], IDENT_AREA                 ;
                01  AB              67          14  28 00038      MOVC3   #20, (IDENT_AREA), 1(R11)               ; 1898
                                    50      04  AC  D0 0003D      MOVL    HEADER, R0                              ; 1899
                                    51      01  A0  9A 00041      MOVZBL  1(R0), R1                               ;
                                    50          60  9A 00045      MOVZBL  (R0), R0                                ;
                    50              51          50  C3 00048      SUBL3   R0, R1, R0                              ;
                                    3C          50  D1 0004C      CMPL    R0, #60                                 ; 1900
                                                08  1F 0004F      BLSSU   1$                                      ;
                15  AB      36      A7  0042    8F  28 00051      MOVC3   #66, 54(IDENT_AREA), 21(R11)            ; 1902
                                                50  D4 00059  1$:  CLRL    J                                       ; 1903
                                    20      01  A04B 91 0005B  2$:  CMPB    1(J)[R11], #32                          ; 1906
                                                0A  13 00060      BEQL    3$                                      ;
                                                6B  96 00062      INCB    (R11)                                   ; 1907
                    EF              50 00000056 8F  F3 00064      AOBLEQ  #86, J, 2$                              ; 1903
                                            18  AA  D5 0006C  3$:  TSTL    24(BASE)                                ; 1918
                                                21  13 0006F      BEQL    5$                                      ;
                        00D4    CA      18  AA  D1 00071      CMPL    24(BASE), 212(BASE)                     ; 1919
                                                19  13 00077      BEQL    5$                                      ;
                                            04  BE  D5 00079      TSTL    @4(SP)                                  ; 1922
                                                0C  13 0007C      BEQL    4$                                      ;
                            01  AA          80  8F  88 0007E      BISB2   #128, 1(BASE)                           ; 1925
                                50          04  BE  D0 00083      MOVL    @4(SP), R0                              ; 1926
                                            18  A0  B6 00087      INCW    24(R0)                                  ;
                                            18  AA  DD 0008A  4$:  PUSHL   24(BASE)                                ; 1929
                        0000G   CF              01  FB 0008D      CALLS   #1, RELEASE_SERIAL_LOCK                 ;
                                56              16  D0 00092  5$:  MOVL    #22, J                                  ; 1932
                                            10  AE  B5 00095  6$:  TSTW    DIR_ID                                  ; 1938
                                                0C  12 00098      BNEQ    7$                                      ;
                                            15  AE  95 0009A      TSTB    DIR_ID+5                                ;
                                                07  12 0009D      BNEQ    7$                                      ;
                                        16          56  D1 0009F      CMPL    J, #22                                  ; 1941
                                                4B  13 000A2      BEQL    11$                                     ;
                                                38  11 000A4      BRB     10$                                     ; 1944
                                        04  10  AE  B1 000A6  7$:  CMPW    DIR_ID, #4                              ; 1950
                                                0A  12 000AA      BNEQ    8$                                      ;
                                            15  AE  95 000AC      TSTB    DIR_ID+5                                ;
                                                05  12 000AF      BNEQ    8$                                      ;
                                        16          56  D1 000B1      CMPL    J, #22                                  ; 1951
                                                39  12 000B4      BNEQ    11$                                     ;
```

```
                    A6   AA          02  8A 000B6 8$:     BICB2   #2, -90(BASE)                                  ; 1959
                                10  AE  9F 000BA          PUSHAB  DIR_ID                                         ; 1960
                 0000G   CF         01  FB 000BD          CALLS   #1, SERIAL_FILE                                ;
                    08   AE         50  D0 000C2          MOVL    R0, TMPINDX                                    ;
                03  A6   AA         01  E1 000C6          BBC     #1, -90(BASE), 9$                              ; 1962
                                08  AE  D4 000CB          CLRL    TMPINDX                                        ; 1964
                                    7E  D4 000CE 9$:      CLRL    -(SP)                                          ; 1966
                                14  AE  9F 000D0          PUSHAB  DIR_ID                                         ;
                 0000G   CF         02  FB 000D3          CALLS   #2, READ_HEADER                                ;
                    0C   AE         50  D0 000D8          MOVL    R0, DIR_HEADER                                 ;
                                    13  12 000DC          BNEQ    12$                                            ; 1968
                50              56  28  C5 000DE 10$:     MULL3   #40, J, R0                                     ; 1971
                              6048  01  90 000E2          MOVB    #1, (R0)[R8]                                   ;
                50              56  28  C5 000E6          MULL3   #40, J, R0                                     ; 1972
                         01  A048  3F  90 000EA          MOVB    #63, 1(R0)[R8]                                 ;
                                    62  11 000EF 11$:     BRB     16$                                            ; 1970
                    50          0C  BE  9A 000F1 12$:     MOVZBL  @DIR_HEADER, R0                                ; 1975
                    57          0C BE40 3E 000F5          MOVAW   @DIR_HEADER[R0], IDENT_AREA                    ;
                50  56              28  C5 000FA          MULL3   #40, J, R0                                     ; 1977
          01 A048   67              14  28 000FE          MOVC3   #20, (IDENT_AREA), 1(R0)[R8]                   ;
                    50          04  AC  D0 00104          MOVL    HEADER, R0                                     ; 1978
                    51          01  A0  9A 00108          MOVZBL  1(R0), R1                                      ;
                    50              60  9A 0010C          MOVZBL  (R0), R0                                       ;
                50  51              50  C3 0010F          SUBL3   R0, R1, R0                                     ;
                    3C              50  D1 00113          CMPL    R0, #60                                        ; 1979
                                        1F 00116          BLSSU   13$                                            ;
                50  56              28  C5 00118          MULL3   #40, J, R0                                     ; 1981
          15 A048   36  A7          13  28 0011C          MOVC3   #19, 54(IDENT_AREA), 21(R0)[R8]                ; 1982
                    51              01  D0 00123 13$:     MOVL    #1, K                                          ; 1983
                50  56              28  C5 00126 14$:     MULL3   #40, J, R0                                     ; 1986
                52  50              51  C1 0012A          ADDL3   K, R0, R2                                      ;
                    2E          6248  91 0012E          CMPB    (R2)[R8], #46                                  ;
                                    07  13 00132          BEQL    15$                                            ;
                                  6048  96 00134          INCB    (R0)[R8]                                       ; 1987
                EB              51      27  F3 00137      AOBLEQ  #39, K, 14$                                    ; 1983
                                04  10  AE  B1 0013B 15$:  CMPW    DIR_ID, #4                                     ; 1994
                                    14  12 0013F          BNEQ    17$                                            ;
                                15  AE  95 00141          TSTB    DIR_ID+5                                       ;
                                    0F  12 00144          BNEQ    17$                                            ;
                                08  AE  D5 00146          TSTL    TMPINDX                                        ; 1997
                                    2D  13 00149          BEQL    20$                                            ;
                                08  AE  DD 0014B          PUSHL   TMPINDX                                        ; 1999
                 0000G   CF         01  FB 0014E          CALLS   #1, RELEASE_SERIAL_LOCK                        ;
                                    23  11 00153 16$:     BRB     20$                                            ; 1996
                7E  0C  AE  00000042    8F  C1 00155 17$:  ADDL3   #66, DIR_HEADER, -(SP)                         ; 2009
          10    AE          9E          06  28 0015E      MOVC3   #6, @(SP)+, DIR_ID                             ;
                                08  AE  D5 00163          TSTL    TMPINDX                                        ; 2011
                                    08  13 00166          BEQL    18$                                            ;
                                08  AE  DD 00168          PUSHL   TMPINDX                                        ; 2013
                 0000G   CF         01  FB 0016B          CALLS   #1, RELEASE_SERIAL_LOCK                        ;
                                    56  F4 00170 18$:     SOBGEQ  J, 19$                                         ; 1932
                                    03  11 00173          BRB     20$                                            ;
                                  FF1D  31 00175 19$:     BRW     6$                                             ;
                                52  90  AA  D0 00178 20$:  MOVL    -112(BASE), R2                                 ; 2019
                                55  1C  A2  D0 0017C      MOVL    28(R2), R5                                     ;
                                        54  D4 00180      CLRL    R4                                             ;
                                51          59  D0 00182 MOVL    R9, R1                                         ;
```

```
                        50              25  D0 00185          MOVL    #37, R0                                         ;
                 00000000G              00  16 00188          JSB     IOC$CVT_DEVNAM                                  ;
                        00    BE        51  A0 0018E          ADDW2   DEVICE_LEN, @0(SP)                              ; 2020
                        50    00    BE  3C 00192          MOVZWL  @0(SP), R0                                      ; 2024
                      6049    5B    8F  90 00196          MOVB    #91, (R0)[R9]                                   ;
                              00    BE  B6 0019B          INCW    @0(SP)                                          ; 2025
                                    57  D4 0019E          CLRL    J                                               ; 2027
                50              57        28  C5 001A0 21$:    MULL3   #40, J, R0                                      ; 2030
                                56      6048  9A 001A4          MOVZBL  (R0)[R8], END_NAME                              ;
                                      1E  13 001A8          BEQL    22$                                             ;
                                52    01    A6  9E 001AA          MOVAB   1(R6), R2                                       ; 2035
                                51    00    BE  3C 001AE          MOVZWL  @0(SP), R1                                      ;
        52      2E      01 A048         56  2C 001B2          MOVC5   END_NAME, 1(R0)[R8], #46, R2, (R1)[R9]          ;
                              6149          001B9                                                                   ;
                                50    00    BE  3C 001BB          MOVZWL  @0(SP), R0                                      ; 2036
                                51    01 A640  9E 001BF          MOVAB   1(END_NAME)[R0], R1                             ;
                              00    BE        51  B0 001C4          MOVW    R1, @0(SP)                                      ;
                        D4              57        16  F3 001C8 22$:    AOBLEQ  #22, J, 21$                             ; 2027
                                50    00    BE  3C 001CC          MOVZWL  @0(SP), R0                                      ; 2042
                              5B    8F        FF A049  91 001D0          CMPB    -1(R0)[R9], #91                         ;
                                            03  12 001D6          BNEQ    23$                                             ;
                                      00    BE  B6 001D8          INCW    @0(SP)                                          ; 2043
                                50    00    BE  3C 001DB 23$:    MOVZWL  @0(SP), R0                                      ; 2045
                              5F A049         5D  8F 90 001DF          MOVB    #93, -1(R0)[R9]                         ;
                                56            6B  9A 001E5          MOVZBL  (R11), END_NAME                         ; 2049
                                50    00    BE  3C 001E8          MOVZWL  @0(SP), R0                                      ; 2050
                      6049      01    AB        56  28 001EC          MOVC3   END_NAME, 1(R11), (R0)[R9]              ;
                              00    BE        56  A0 001F2          ADDW2   END_NAME, @0(SP)                                ; 2051
                                      18  AA  D5 001F6          TSTL    24(BASE)                                        ; 2056
                                            34  13 001F9          BEQL    27$                                             ;
                                      04    BE  D5 001FB          TSTL    @4(SP)                                          ; 2059
                                            07  13 001FE          BEQL    24$                                             ;
                50              04    BE        24  C1 00200          ADDL3   #36, @4(SP), R0                                 ; 2060
                                            05  11 00205          BRB     25$                                             ; 2061
                50              10    AA        04  C1 00207 24$:    ADDL3   #4, 16(BASE), R0                        ;
                                            50  DD 0020C 25$:    PUSHL   R0                                      ;
                              0000G   CF        01  FB 0020E          CALLS   #1, SERIAL_FILE                         ;
                                18    AA        50  D0 00213          MOVL    R0, 24(BASE)                            ;
                07                    6A        0F  E5 00217          BBCC    #15, (BASE), 26$                        ; 2063
                                      50  04    BE  D0 0021B          MOVL    @4(SP), R0                              ; 2065
                                            18  A0  B7 0021F          DECW    24(R0)                                  ;
                                      04    BE  DD 00222 26$:    PUSHL   @4(SP)                                  ; 2067
                7E              10    AA        04  C1 00225          ADDL3   #4, 16(BASE), -(SP)                     ;
                              0000G   CF        02  FB 0022A          CALLS   #2, READ_HEADER                         ;
                                                  04 0022F 27$:    RET                                             ; 2072

; Routine Size:  560 bytes,    Routine Base:  $CODE$ + 0556
```

```
;1087     2073  1 ROUTINE READ_HANDLER (SIG_ARGS, MECH_ARGS) : NOVALUE =
;1088     2074  1
;1089     2075  1 !++
;1090     2076  1 !
;1091     2077  1 ! FUNCTIONAL DESCRIPTION:
;1092     2078  1 !
;1093     2079  1 !       This routine is a condition handler for READ_ATTRIB. It catches
;1094     2080  1 !       error exits from MAKE_NAMEBLOCK (due to garbage in the file header
;1095     2081  1 !       name string) and causes them to be ignored.
;1096     2082  1 !
;1097     2083  1 ! CALLING SEQUENCE:
;1098     2084  1 !       READ_HANDLER (ARG1, ARG2)
;1099     2085  1 !
;1100     2086  1 ! INPUT PARAMETERS:
;1101     2087  1 !       ARG1: address of signal array
;1102     2088  1 !       ARG2: address of mechanism array
;1103     2089  1 !
;1104     2090  1 ! IMPLICIT INPUTS:
;1105     2091  1 !       NONE
;1106     2092  1 !
;1107     2093  1 ! OUTPUT PARAMETERS:
;1108     2094  1 !       NONE
;1109     2095  1 !
;1110     2096  1 ! IMPLICIT OUTPUTS:
;1111     2097  1 !       NONE
;1112     2098  1 !
;1113     2099  1 ! ROUTINE VALUE:
;1114     2100  1 !       NONE
;1115     2101  1 !
;1116     2102  1 ! SIDE EFFECTS:
;1117     2103  1 !       stack unwound to return to call site in READ_ATTRIB
;1118     2104  1 !
;1119     2105  1 !--
;1120     2106  1
;1121     2107  2 BEGIN
;1122     2108  2
;1123     2109  2 MAP
;1124     2110  2         SIG_ARGS                : REF BBLOCK,   ! signal array arg
;1125     2111  2         MECH_ARGS       : REF BBLOCK;   ! mechanism array arg
;1126     2112  2
;1127     2113  2
;1128     2114  2 ! Check the signal code. The only permissible ones are SS$_UNWIND, which
;1129     2115  2 ! is ignored, and SS$_CMODUSER. The error status is the 16 bit CHMU code.
;1130     2116  2 !
;1131     2117  2
;1132     2118  2 IF .SIG_ARGS[CHF$L_SIG_NAME] EQL SS$_UNWIND THEN RETURN;
;1133     2119  2 IF .SIG_ARGS[CHF$L_SIG_NAME] NEQ SS$_CMODUSER
;1134     2120  2 THEN BUG_CHECK (UNXSIGNAL, FATAL, 'Unexpected signal name in ACP');
;1135     2121  2
;1136     2122  2 $UNWIND (DEPADR = MECH_ARGS[CHF$L_MCH_DEPTH]);
;1137     2123  2
;1138     2124  2 RETURN;
;1139     2125  2
;1140     2126  1 END;                                    ! end of routine READ_HANDLER
```

```
                                                      .EXTRN  BUG$_UNXSIGNAL, SYS$UNWIND

                                0000 00000 READ_HANDLER:
                                                      .WORD   Save nothing                        ; 2073
                 50         04    AC  D0 00002        MOVL    SIG_ARGS, R0                        ; 2118
       00000920  8F         04    A0  D1 00006        CMPL    4(R0), #2336                        ;
                                  1C  13 0000E        BEQL    2$                                  ;
       00000424  8F         04    A0  D1 00010        CMPL    4(R0), #1060                        ; 2119
                                  04  13 00018        BEQL    1$                                  ;
                                    FEFF 0001A        BUGW                                        ; 2120
                                   0000* 0001C        .WORD   <BUG$_UNXSIGNAL!4>                  ;
                                  7E  D4 0001E 1$:    CLRL    -(SP)                               ; 2122
    7E         08    AC           08  C1 00020        ADDL3   #8, MECH_ARGS, -(SP)                ;
       00000000G     00           02  FB 00025        CALLS   #2, SYS$UNWIND                      ;
                                      04 0002C 2$:    RET                                         ; 2126

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0786
```

```
;  1142          2127  1 GLOBAL ROUTINE WRITE_ATTRIB (HEADER, ABD, CONTROL_ACCESS) : L_NORM NOVALUE =
;  1143          2128  1
;  1144          2129  1 !++
;  1145          2130  1 !
;  1146          2131  1 ! FUNCTIONAL DESCRIPTION:
;  1147          2132  1 !
;  1148          2133  1 !       This routine performs the write attributes function. The
;  1149          2134  1 !       requested attributes are taken from the buffer packet.
;  1150          2135  1 !
;  1151          2136  1 ! CALLING SEQUENCE:
;  1152          2137  1 !       READ_ATTRIB (ARG1, ARG2, ARG3)
;  1153          2138  1 !
;  1154          2139  1 ! INPUT PARAMETERS:
;  1155          2140  1 !       ARG1: address of file header
;  1156          2141  1 !       ARG2: address of buffer descriptors
;  1157          2142  1 !       ARG3: 1 = check for control access to the file
;  1158          2143  1 !             0 = no control access check
;  1159          2144  1 !
;  1160          2145  1 ! IMPLICIT INPUTS:
;  1161          2146  1 !       IO_PACKET: I/O packet for this operation
;  1162          2147  1 !       PRIMARY_FCB: FCB of file
;  1163          2148  1 !
;  1164          2149  1 ! OUTPUT PARAMETERS:
;  1165          2150  1 !       NONE
;  1166          2151  1 !
;  1167          2152  1 ! IMPLICIT OUTPUTS:
;  1168          2153  1 !       NONE
;  1169          2154  1 !
;  1170          2155  1 ! ROUTINE VALUE:
;  1171          2156  1 !       NONE
;  1172          2157  1 !
;  1173          2158  1 ! SIDE EFFECTS:
;  1174          2159  1 !       attribute data written into appropriate places
;  1175          2160  1 !
;  1176          2161  1 !--
;  1177          2162  1
;  1178          2163  2 BEGIN
;  1179          2164  2
;  1180          2165  2 MAP
;  1181          2166  2         HEADER          : REF BBLOCK,   ! file header arg
;  1182          2167  2         ABD             : REF BBLOCKVECTOR [.ABD$C_LENGTH];
;  1183          2168  2                                         ! buffer descriptor arg
;  1184          2169  2
;  1185          2170  2 LOCAL
;  1186          2171  2         LOCAL_HEADER    : REF BBLOCK,   ! Local copy of the header address
;  1187          2172  2         CTL_ACC_GRANTED,                ! Flag indicating control access granted
;  1188          2173  2         ACL_MODIFIED,                   ! flag indicating ACL has been modified
;  1189          2174  2         SAVE_HIBLK,                     ! saved copy of file's HIBLK
;  1190          2175  2         SAVE_CHAR       : BBLOCK [4],   ! initial state of protected attributes
;  1191          2176  2         STATUS;                         ! routine exit status
;  1192          2177  2
;  1193          2178  2 BIND_COMMON;
;  1194          2179  2
;  1195          2180  2 EXTERNAL ROUTINE
;  1196          2181  2         PMS_START_SUB   : L_NORM,       ! start subfunction metering
;  1197          2182  2         PMS_END_SUB     : L_NORM,       ! end subfunction metering
;  1198          2183  2         CHECKSUM        : L_NORM,       ! Checksum the header
```

```
;  1199          2184  2          MARK_DIRTY      : L_NORM,          ! mark buffer for write-back
;  1200          2185  2          UPDATE_DIRSEQ   : L_NORM,          ! update UCB directory sequence count
;  1201          2186  2          KILL_BUFFERS    : L_NORM,          ! flush a file from the buffer cache
;  1202          2187  2          KILL_DINDX      : L_NORM NOVALUE,  ! kill directory index
;  1203          2188  2          MAKE_NAMEBLOCK  : L_NORM,          ! convert file string to RAD-50 name block
;  1204          2189  2          MAKE_STRING     : L_NORM,          ! convert name block into file string
;  1205          2190  2          READ_HEADER     : L_NORM,          ! read a file header
;  1206          2191  2          ACL_DISPATCH    : L_NORM,          ! ACL action dispatcher
;  1207          2192  2          ACL_BUILDACL    : L_NORM,          ! rebuild ACL in file header
;  1208          2193  2          CHECK_PROTECT   : L_NORM;          ! Protection checking routine
;  1209          2194  2
;  1210          2195  2
;  1211          2196  2  ! Start metering for this subfunction.
;  1212          2197  2  !
;  1213          2198  2
;  1214          2199  2  PMS_START_SUB (PMS_RWATT);
;  1215          2200  2
;  1216          2201  2  LOCAL_HEADER = .HEADER;
;  1217          2202  2  CTL_ACC_GRANTED = 0;
;  1218          2203  2  ACL_MODIFIED = 0;
;  1219          2204  2  STATUS = SS$_NORMAL;                          ! Assume success
;  1220          2205  2
;  1221          2206  2  ! Set the appropriate cleanup flags and save the initial state of the
;  1222          2207  2  ! protected file characteristics.
;  1223          2208  2  !
;  1224          2209  2
;  1225          2210  2  CHECKSUM (.LOCAL_HEADER);
;  1226          2211  2  MARK_DIRTY (.LOCAL_HEADER);
;  1227          2212  2  CLEANUP_FLAGS[CLF_FIXFCB] = 1;
;  1228          2213  2
;  1229          2214  2  SAVE_CHAR = .LOCAL_HEADER[FH2$L_FILECHAR];
;  1230          2215  2  SAVE_HIBLK = .BBLOCK [LOCAL_HEADER[FH2$W_RECATTR], FAT$L_HIBLK];
;  1231          2216  2
;  1232          2217  2
;  1233          2218  2  ! Scan the buffer packet, picking up each entry. The first byte of the
;  1234          2219  2  ! text is the attribute code.
;  1235          2220  2  !
;  1236          2221  2
;  1237          2222  2  INCR I FROM ABD$C_ATTRIB TO .IO_PACKET[IRP$W_BCNT]-1 DO
;  1238          2223  3      BEGIN
;  1239          2224  3
;  1240          2225  3      LOCAL
;  1241          2226  3          P,                                 ! pointer to attribute text
;  1242          2227  3          T,                                 ! temporary pointer
;  1243          2228  3          COUNT,                             ! attribute size desired
;  1244          2229  3          ADDRESS         : REF BBLOCK,      ! address of attribute
;  1245          2230  3          CODE,                              ! attribute code
;  1246          2231  3          MAX_COUNT,                         ! max size of attribute
;  1247          2232  3          ACTION          : BYTE,            ! code of action routine
;  1248          2233  3          ATT_BUFFER      : BBLOCK [44];     ! attribute copy buffer
;  1249          2234  3
;  1250          2235  3      P = .ABD[.I, ABD$W_TEXT] + ABD[.I, ABD$W_TEXT];
;  1251          2236  3      COUNT = .ABD[.I, ABD$W_COUNT];
;  1252          2237  3      CODE = .(.P)<0,8> - 1;
;  1253          2238  3      P = .P + 1;
;  1254          2239  3
;  1255          2240  3  ! Check the attribute code for legality, and then check the requested
```

```
;  1256          2241  3 ! size against the limit.
;  1257          2242  3 !
;  1258          2243  3
;  1259          2244  3     IF .CODE GTR MAX_CODE - 1
;  1260          2245  3     THEN (STATUS = SS$_BADATTRIB; EXITLOOP);
;  1261          2246  3
;  1262          2247  3     MAX_COUNT = .ATC[.CODE, ATC_MAX_SIZE];
;  1263          2248  3     IF .COUNT GTR .MAX_COUNT
;  1264          2249  3     THEN (STATUS = SS$_BADATTRIB; EXITLOOP);
;  1265          2250  3
;  1266          2251  3     IF .ATC[.CODE, ATC_PROTECTED]
;  1267          2252  3     THEN
;  1268          2253  4         BEGIN
;  1269          2254  4
;  1270          2255  4 ! Most of the protected fields will affect the protection checking, so
;  1271          2256  4 ! mark other fcbs as stale to cause them to reconstruct their fcbs from
;  1272          2257  4 ! the header.
;  1273          2258  4 !
;  1274          2259  4
;  1275          2260  4         CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1276          2261  4
;  1277          2262  4         IF .CONTROL_ACCESS AND NOT .CTL_ACC_GRANTED
;  1278          2263  4         THEN
;  1279          2264  5             BEGIN
;  1280          2265  5
;  1281          2266  5 ! Restore protected field in case the protection check fails.
;  1282          2267  5 !
;  1283          2268  5
;  1284          2269  5             BBLOCK [LOCAL_HEADER[FH2$W_RECATTR], FAT$L_HIBLK] = .SAVE_HIBLK;
;  1285          2270  5             CHECK_PROTECT (WRATT_ACCESS, .LOCAL_HEADER, .PRIMARY_FCB,
;  1286          2271  5                            MAXU (.IO_PACKET[IRP$V_MODE], .CURRENT_FIB[FIB$B_AGENT_MODE]));
;  1287          2272  5
;  1288          2273  5 ! Control access is allowed.  Note this for future use, to avoid additional
;  1289          2274  5 ! (unneeded) protection checks.
;  1290          2275  5 !
;  1291          2276  5
;  1292          2277  5             CTL_ACC_GRANTED = 1;
;  1293          2278  4             END;
;  1294          2279  3         END;
;  1295          2280  3
;  1296          2281  3
;  1297          2282  3 ! Compute the action routine code.
;  1298          2283  3 !
;  1299          2284  3
;  1300          2285  3     ACTION = .ATC[.CODE, ATC_ACTION];
;  1301          2286  3     IF .ATC[.CODE, ATC_READ_ONLY] AND .ACTION NEQ ACT_ACL
;  1302          2287  3     THEN ACTION = ACT_NOP;
;  1303          2288  3
;  1304          2289  3 ! Compute the address of the attribute.
;  1305          2290  3 !
;  1306          2291  3
;  1307          2292  3     ADDRESS =
;  1308          2293  4         (
;  1309          2294  4         CASE .ATC[.CODE, ATC_LOCATION] FROM 0 TO ATC_LASTATC OF
;  1310          2295  4             SET
;  1311          2296  4             [ATC_ZERO,
;  1312          2297  4              ATC_ACPGBL,
```

```
;  1313     2298  4                ATC_FID2NAME]:      ATT_BUFFER;
;  1314     2299  4                [ATC_FCB]:          .PRIMARY_FCB;
;  1315     2300  5                [ATC_HEADER]:       BEGIN
;  1316     2301  5                                    IF .ATC[.CODE, ATC_OFFSET]
;  1317     2302  5                                     + .ATC[.CODE, ATC_DATA_SIZE] GTRU
;  1318     2303  5                                       .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  1319     2304  5                                    THEN ACTION = ACT_ZERO;
;  1320     2305  5                                    .LOCAL_HEADER
;  1321     2306  4                                    END;
;  1322     2307  5                [ATC_IDENT]·        BEGIN
;  1323     2308  5                                    IF .ATC[.CODE, ATC_OFFSET]
;  1324     2309  5                                     + .ATC[.CODE, ATC_DATA_SIZE] GTRU
;  1325     2310  5                                       .LOCAL_HEADER[FH2$B_MPOFFSET]*2
;  1326     2311  5                                     - .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  1327     2312  5                                    THEN ACTION = ACT_ZERO;
;  1328     2313  5                                    .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_IDOFFSET]*2
;  1329     2314  4                                    END;
;  1330     2315  4                [ATC_MAP]:          .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_MPOFFSET]*2;
;  1331     2316  4                [ATC_ACL]:          .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_ACOFFSET]*2;
;  1332     2317  4                [ATC_RESERVED]:     .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_RSOFFSET]*2;
;  1333     2318  4                TES
;  1334     2319  4            )
;  1335     2320  3            + .ATC[.CODE, ATC_OFFSET];
;  1336     2321  3
;  1337     2322  3
;  1338     2323  3    ! Finally execute the action routine.
;  1339     2324  3    !
;  1340     2325  3
;  1341     2326  3        CASE .ACTION FROM 0 TO ACT_LASTACT OF
;  1342     2327  3            SET
;  1343     2328  3
;  1344     2329  3            [ACT_NOP,
;  1345     2330  3             ACT_BLOCKSIZE,
;  1346     2331  3             ACT_ZERO,
;  1347     2332  3             ACT_BLANK,
;  1348     2333  3             ACT_ACMODE,
;  1349     2334  3             ACT_STATBLK]:  0;
;  1350     2335  3
;  1351     2336  3            [ACT_ILLEGAL]:  (STATUS = SS$_BADATTRIB; EXITLOOP);
;  1352     2337  3
;  1353     2338  4            [ACT_UIC2]:     BEGIN
;  1354     2339  4                            LOCAL UIC;
;  1355     2340  4                            UIC = ..ADDRESS;
;  1356     2341  4                            UIC<0,16> = .(.P)<0,8>;
;  1357     2342  4                            IF .COUNT GEQ 2
;  1358     2343  4                            THEN UIC<16,16> = .(.P+1)<0,8>;
;  1359     2344  4                            STATUS = CHANGE_OWNER (.UIC, .PRIMARY_FCB, .LOCAL_HEADER);
;  1360     2345  4                            IF NOT .STATUS THEN EXITLOOP;
;  1361     2346  4                            LOCAL_HEADER = .FILE_HEADER;
;  1362     2347  4                            IF .COUNT GEQ 3
;  1363     2348  4                            THEN (LOCAL_HEADER[FH2$W_FILEPROT])<0,8> = .(.P+2)<0,8>;
;  1364     2349  4                            IF .COUNT GEQ 4
;  1365     2350  4                            THEN (LOCAL_HEADER[FH2$W_FILEPROT])<8,8> = .(.P+3)<0,8>;
;  1366     2351  4                            IF .COUNT GEQ 5
;  1367     2352  4                            THEN (LOCAL_HEADER[FH2$L_FILECHAR])<0,8> = .(.P+4)<0,8>;
;  1368     2353  4                            CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1369     2354  3                            END;
```

```
;  1370     2355  3
;  1371     2356  4          [ACT_UIC4]:     BEGIN
;  1372     2357  4                          LOCAL UIC;
;  1373     2358  4                          UIC = ..ADDRESS;
;  1374     2359  4                          CH$MOVE (.COUNT, .P, UIC);
;  1375     2360  4                          STATUS = CHANGE_OWNER (.UIC, .PRIMARY_FCB, .LOCAL_HEADER);
;  1376     2361  4                          LOCAL_HEADER = .FILE_HEADER;
;  1377     2362  4                          IF NOT .STATUS THEN EXITLOOP;
;  1378     2363  4                          CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1379     2364  3                          END;
;  1380     2365  3
;  1381     2366  4          [ACT_CLASS]:    BEGIN
;  1382     2367  4                          LOCAL   CLASS_BLOCK     : BBLOCK [ATR$S_CLASS_MASK];
;  1383     2368  4                          CH$COPY (.COUNT, .P, 0, ATR$S_CLASS_MASK, CLASS_BLOCK);
;  1384     2369  4                          STATUS = CHANGE_CLASS (.LOCAL_HEADER, CLASS_BLOCK);
;  1385     2370  4                          IF NOT .STATUS THEN EXITLOOP;
;  1386     2371  4                          CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1387     2372  3                          END;
;  1388     2373  3
;  1389     2374  4          [ACT_FPRO]:     BEGIN
;  1390     2375  4                          (LOCAL_HEADER[FH2$W_FILEPROT])<0,8> = .(.P+0)<0,8>;
;  1391     2376  4                          IF .COUNT GEQ 2
;  1392     2377  4                          THEN (LOCAL_HEADER[FH2$W_FILEPROT])<8,8> = .(.P+1)<0,8>;
;  1393     2378  4                          IF .COUNT GEQ 3
;  1394     2379  4                          THEN (LOCAL_HEADER[FH2$L_FILECHAR])<0,8> = .(.P+2)<0,8>;
;  1395     2380  4                          CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1396     2381  3                          END;
;  1397     2382  3
;  1398     2383  4          [ACT_ACLEVEL]:  BEGIN
;  1399     2384  4                          LOCAL MODE;
;  1400     2385  4                          (.ADDRESS)<0,8> = .(.P)<0,8>;
;  1401     2386  4                          MODE = NOT MAXU (.IO_PACKET[IRP$V_MODE], .CURRENT_FIB[FIB$B_AGENT_MODE]) AND 3;
;  1402     2387  4                          INCR J FROM 0 TO 3
;  1403     2388  4                          DO
;  1404     2389  5                              BEGIN
;  1405     2390  5                              IF .(.ADDRESS)<.J*2,2> GTRU .MODE
;  1406     2391  5                              THEN (.ADDRESS)<.J*2,2> = .MODE;
;  1407     2392  4                              END;
;  1408     2393  3                          END;
;  1409     2394  3
;  1410     2395  4          [ACT_FILENAME]: BEGIN
;  1411     2396  4                          CH$COPY (.COUNT, .P, ' ',
;  1412     2397  4                          FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME]);
;  1413     2398  4                          IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  1414     2399  4                          GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  1415     2400  4                          THEN CH$COPY (MAX (.COUNT-FI2$S_FILENAME, 0), .P+FI2$S_FILENAME, ' ',
;  1416     2401  4                          FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT]);
;  1417     2402  3                          END;
;  1418     2403  3
;  1419     2404  4          [ACT_R50_NAME]: BEGIN
;  1420     2405  4                          T = ATT_BUFFER[NMB$W_NAME];
;  1421     2406  4                          IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  1422     2407  4                          GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  1423     2408  4                          THEN CH$FILL (' ', FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT]);
;  1424     2409  4                          MAKE_NAMEBLOCK (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ATT_BUFFER);
;  1425     2410  4                          CH$MOVE (.COUNT, .P, .T);
;  1426     2411  4                          CH$FILL (' ', FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME]);
```

```
;  1427          2412  4                    MAKE_STRING (ATT_BUFFER, ADDRESS[FI2$T_FILENAME]);
;  1428          2413  3                    END;
;  1429          2414  3
;  1430          2415  4            [ACT_R50_TYPE]: BEGIN
;  1431          2416  4                    T = ATT_BUFFER[NMB$W_TYPE];
;  1432          2417  4                    IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  1433          2418  4                    GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  1434          2419  4                    THEN CH$FILL (' ', FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT]);
;  1435          2420  4                    MAKE_NAMEBLOCK (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ATT_BUFFER);
;  1436          2421  4                    CH$MOVE (.COUNT, .P, .T);
;  1437          2422  4                    CH$FILL (' ', FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME]);
;  1438          2423  4                    MAKE_STRING (ATT_BUFFER, ADDRESS[FI2$T_FILENAME]);
;  1439          2424  3                    END;
;  1440          2425  3
;  1441          2426  4            [ACT_R50_VER]:  BEGIN
;  1442          2427  4                    T = ATT_BUFFER[NMB$W_VERSION];
;  1443          2428  4                    IF .LOCAL_HEADER[FH2$B_MPOFFSET] - .LOCAL_HEADER[FH2$B_IDOFFSET]
;  1444          2429  '                    GEQU ($BYTEOFFSET (FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT) / 2
;  1445          2430  4                    THEN CH$FILL (' ', FI2$S_FILENAMEXT, ADDRESS[FI2$T_FILENAMEXT]);
;  1446          2431  4                    MAKE_NAMEBLOCK (FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME], ATT_BUFFER);
;  1447          2432  4                    CH$MOVE (.COUNT, .P, .T);
;  1448          2433  4                    CH$FILL (' ', FI2$S_FILENAME, ADDRESS[FI2$T_FILENAME]);
;  1449          2434  4                    MAKE_STRING (ATT_BUFFER, ADDRESS[FI2$T_FILENAME]);
;  1450          2435  3                    END;
;  1451          2436  3
;  1452          2437  4            [ACT_DATE]:     BEGIN
;  1453          2438  4                    CH$COPY (.COUNT, .P, '0', 13, ATT_BUFFER);
;  1454          2439  4                    CONVERT_DATE (ATT_BUFFER, .ADDRESS);
;  1455          2440  3                    END;
;  1456          2441  3
;  1457          2442  4            [ACT_DATES]:    BEGIN
;  1458          2443  4                    CH$COPY (.COUNT, .P, '0', 44, ATT_BUFFER);
;  1459          2444  4                    ADDRESS[FI2$W_REVISION] = .ATT_BUFFER;
;  1460          2445  4                    CONVERT_DATE (ATT_BUFFER+02, ADDRESS[FI2$Q_REVDATE]);
;  1461          2446  4                    CONVERT_DATE (ATT_BUFFER+15, ADDRESS[FI2$Q_CREDATE]);
;  1462          2447  4                    CONVERT_DATE (ATT_BUFFER+28, ADDRESS[FI2$Q_EXPDATE]);
;  1463          2448  3                    END;
;  1464          2449  3
;  1465          2450  3            [ACT_COPY]:     CH$MOVE (.COUNT, .P, .ADDRESS);
;  1466          2451  3
;  1467          2452  4            [ACT_ACL]:      BEGIN
;  1468          2453  4                    IF .CURRENT_FIB[FIB$L_ACL_STATUS]
;  1469          2454  4                    THEN
;  1470          2455  5                        BEGIN
;  1471          2456  5                        CHECKSUM (.LOCAL_HEADER);
;  1472          2457  5                        MARK_DIRTY (.LOCAL_HEADER);
;  1473          2458  5                        ACL_MODIFIED = 1;
;  1474        P 2459  5                        STATUS = KERNEL_CALL (ACL_DISPATCH, .CODE,
;  1475          2460  5                                              .ADDRESS, .COUNT, .P);
;  1476          2461  5                        LOCAL_HEADER = .FILE_HEADER;
;  1477          2462  5                        IF NOT .STATUS
;  1478          2463  5                        THEN LOCAL_FIB[FIB$L_ACL_STATUS] = .STATUS;
;  1479          2464  4                        END;
;  1480          2465  4                    STATUS = 1;
;  1481          2466  4                    CLEANUP_FLAGS [CLF_MARKFCBSTALE] = 1;
;  1482          2467  3                    END;
;  1483          2468  3
```

```
; 1484        2469  4          [ACT_RESERVED]: BEGIN
; 1485        2470  4                          LOCAL   MAP_END,
; 1486        2471  4                                  ACL_END : REF BBLOCK,
; 1487        2472  4                                  RESERVED_LENGTH;
; 1488        2473  4
; 1489        2474  4                          IF .COUNT LSS 2 THEN (STATUS = SS$_BADATTRIB; EXITLOOP);
; 1490        2475  4
; 1491        2476  4 ! Determine where the maximum length possible for the reserved area.
; 1492        2477  4 !
; 1493        2478  4
; 1494        2479  4                          MAP_END = .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_MPOFFSET]*2 +
; 1495        2480  4                                    .LOCAL_HEADER[FH2$B_MAP_INUSE]*2;
; 1496        2481  4                          ACL_END = .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_ACOFFSET]*2;
; 1497        2482  4                          UNTIL .ACL_END GEQ .LOCAL_HEADER + .LOCAL_HEADER[FH2$B_RSOFFSET]*2
; 1498        2483  4                          DO
; 1499        2484  5                              BEGIN
; 1500        2485  5                              IF .ACL_END[ACE$B_SIZE] EQL 0 THEN EXITLOOP;
; 1501        2486  5                              ACL_END = .ACL_END + .ACL_END[ACE$B_SIZE];
; 1502        2487  4                              END;
; 1503        2488  4                          RESERVED_LENGTH = LOCAL_HEADER[FH2$W_CHECKSUM] -
; 1504        2489  5                                            (IF .LOCAL_HEADER[FH2$B_ACOFFSET] NEQ
; 1505        2490  5                                                .LOCAL_HEADER[FH2$B_RSOFFSET]
; 1506        2491  4                                             THEN .ACL_END ELSE .MAP_END);
; 1507        2492  4
; 1508        2493  4                          CH$FILL (0, $BYTEOFFSET (FH2$W_CHECKSUM) -
; 1509        2494  4                                      .RESERVED_LENGTH,
; 1510        2495  4                                     LOCAL_HEADER[FH2$W_CHECKSUM] - .RESERVED_LENGTH);
; 1511        2496  4                          P = .P + 2;                 ! Skip over size
; 1512        2497  4                          COUNT = .COUNT - 2;
; 1513        2498  4                          IF .COUNT GTR .RESERVED_LENGTH THEN (STATUS = SS$_HEADERFULL; EXITLOOP);
; 1514        2499  4                          ADDRESS = (LOCAL_HEADER[FH2$W_CHECKSUM] - .COUNT) AND NOT 1;
; 1515        2500  4                          IF .LOCAL_HEADER[FH2$B_ACOFFSET] EQL .LOCAL_HEADER[FH2$B_RSOFFSET]
; 1516        2501  4                          AND .LOCAL_HEADER[FH2$B_ACOFFSET] NEQ .LOCAL_HEADER[FH2$B_MPOFFSET] + .LOCAL_HEADER[
; 1517        2502  4                          THEN LOCAL_HEADER[FH2$B_ACOFFSET] =
; 1518        2503  4                               LOCAL_HEADER[FH2$B_RSOFFSET] = (.ADDRESS - .LOCAL_HEADER) / 2
; 1519        2504  4                          ELSE LOCAL_HEADER[FH2$B_RSOFFSET] = (.ADDRESS - .LOCAL_HEADER) / 2;
; 1520        2505  4                          CH$MOVE (.COUNT, .P, .ADDRESS);
; 1521        2506  3                          END;
; 1522        2507  3
; 1523        2508  3                      TES;
; 1524        2509  3
; 1525        2510  2                  END;                                ! end of loop
; 1526        2511  2
; 1527        2512  2 ! If the directory bit was turned off by this operation, we must purge
; 1528        2513  2 ! the RMS directory caches and the directory block cache on this volume.
; 1529        2514  2 !
; 1530        2515  2
; 1531        2516  2 IF .SAVE_CHAR[FCH$V_DIRECTORY]
; 1532        2517  2 AND NOT .LOCAL_HEADER[FH2$V_DIRECTORY]
; 1533        2518  2 THEN
; 1534        2519  3     BEGIN
; 1535        2520  3     UPDATE_DIRSEQ ();
; 1536        2521  3     KILL_BUFFERS (1, .LB_BASIS [.PRIM_LCKINDX]);
; 1537        2522  3
; 1538        2523  3     IF .PRIMARY_FCB [FCB$L_DIRINDX] NEQ 0
; 1539        2524  3     THEN
; 1540        2525  3         KILL_DINDX (.PRIMARY_FCB);
```

```
;  1541          2526  3
;  1542          2527  2      END;
;  1543          2528  2
;  1544          2529  2  ! Restore the state of the protected file characteristics bits.
;  1545          2530  2  !
;  1546          2531  2
;  1547          2532  3  LOCAL_HEADER[FH2$L_FILECHAR] = (.HEADER[FH2$L_FILECHAR] AND NOT PROTECTED_CHAR)
;  1548          2533  2                                 OR (.SAVE_CHAR AND PROTECTED_CHAR);
;  1549          2534  2  BBLOCK [LOCAL_HEADER[FH2$W_RECATTR], FAT$L_HIBLK] = .SAVE_HIBLK;
;  1550          2535  2
;  1551          2536  2  ! Rebuild the header's ACL if necessary. Otherwise just checksum it.
;  1552          2537  2  !
;  1553          2538  2
;  1554          2539  2  IF .ACL_MODIFIED
;  1555          2540  2  THEN
;  1556          2541  2      ACL_BUILDACL (.PRIMARY_FCB)
;  1557          2542  2  ELSE
;  1558          2543  3      BEGIN
;  1559          2544  3      CHECKSUM (.LOCAL_HEADER);
;  1560          2545  3      MARK_DIRTY (.LOCAL_HEADER);
;  1561          2546  2      END;
;  1562          2547  2
;  1563          2548  2  ! Stop metering of this subfunction
;  1564          2549  2  !
;  1565          2550  2
;  1566          2551  2  PMS_END_SUB ();
;  1567          2552  2
;  1568          2553  2  IF NOT .STATUS THEN ERR_EXIT (.STATUS);
;  1569          2554  2
;  1570          2555  1  END;                                    ! end of routine WRITE_ATTRIB


                                                .EXTRN  CHECKSUM, MARK_DIRTY
                                                .EXTRN  UPDATE_DIRSEQ, KILL_BUFFERS
                                                .EXTRN  KILL_DINDX, MAKE_STRING
                                                .EXTRN  ACL_BUILDACL, CHECK_PROTECT

                                    0BFC 00000  .ENTRY  WRITE_ATTRIB, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 2127
                                                        R11                                           ;
                    5E        94    AE   9E 00002     MOVAB   -108(SP), SP                          ;
                20  AE        08    AA   9E 00006     MOVAB   8(BASE), 32(SP)                       ; 2176
                1C  AE      0204    CA   9E 0000B     MOVAB   516(BASE), 28(SP)                     ;
                                    09   DD 00011     PUSHL   #9                                    ; 2199
            0000G   CF              01   FB 00013     CALLS   #1, PMS_START_SUB                     ;
                    56        04    AC   D0 00018     MOVL    HEADER, LOCAL_HEADER                  ; 2201
                              14    AE   7C 0001C     CLRQ    ACL_MODIFIED                          ; 2203
                10  AE              01   D0 0001F     MOVL    #1, STATUS                            ; 2204
                                    56   DD 00023     PUSHL   LOCAL_HEADER                          ; 2210
            0000G   CF              01   FB 00025     CALLS   #1, CHECKSUM                          ;
                                    56   DD 0002A     PUSHL   LOCAL_HEADER                          ; 2211
            0000G   CF              01   FB 0002C     CALLS   #1, MARK_DIRTY                        ;
                    6A              02   88 00031     BISB2   #2, (BASE)                            ; 2212
                    5B        34    A6   D0 00034     MOVL    52(LOCAL_HEADER), SAVE_CHAR           ; 2214
                0C  AE        18    A6   D0 00038     MOVL    24(LOCAL_HEADER), SAVE_HIBLK          ; 2215
                    50        90    AA   D0 0003D     MOVL    -112(BASE), R0                        ; 2222
                08  AE        32    A0   3C 00041     MOVZWL  50(R0), 8(SP)                         ;
```

```
                         04    AE         04  D0 00046       MOVL    #4, I                                     ;
                                        0415  31 0004A       BRW     64$                                       ;
                               51    04   AE  D0 0004D 1$:   MOVL    I, R1                                     ; 2235
                               50    08 BC41  7E 00051       MOVAQ   @ABD[R1], R0                              ;
                               51         60  3C 00056       MOVZWL  (R0), R1                                  ;
                    6E         51         50  C1 00059       ADDL3   R0, R1, P                                 ;
                               58    02   A0  3C 0005D       MOVZWL  2(R0), COUNT                              ; 2236
                               59    00   BE  9A 00061       MOVZBL  @P, CODE                                  ; 2237
                                          59  D7 00065       DECL    CODE                                      ;
                                          6E  D6 00067       INCL    P                                         ; 2238
                    2F                    59  D1 00069       CMPL    CODE, #47                                 ; 2244
                                          0B  14 0006C       BGTR    2$                                        ;
                                  F7E0 CF49  7F 0006E        PUSHAQ  ATC+6[CODE]                               ; 2247
                               50         9E  3C 00073       MOVZWL  @(SP)+, MAX_COUNT                         ;
                               50         58  D1 00076       CMPL    COUNT, MAX_COUNT                          ; 2248
                                          03  15 00079 2$:   BLEQ    3$                                        ;
                                        0330  31 0007B       BRW     55$                                       ;
                                  F7CA CF49  7F 0007E 3$:    PUSHAQ  ATC[CODE]                                 ; 2251
                    3A         9E         01  E1 00083       BBC     #1, @(SP)+, 5$                            ;
                         01    AA    40   8F  88 00087       BISB2   #64, 1(BASE)                              ; 2260
                               31    0C   AC  E9 0008C       BLBC    CONTROL_ACCESS, 5$                        ; 2262
                               2D    18   AE  E8 00090       BLBS    CTL_ACC_GRANTED, 5$                       ;
                         18    A6    0C   AE  D0 00094       MOVL    SAVE_HIBLK, 24(LOCAL_HEADER)              ; 2269
                               51    90   AA  D0 00099       MOVL    -112(BASE), R1                            ; 2271
                               50    10   AA  D0 0009D       MOVL    16(BASE), R0                              ;
        7E         0B    A1               02  00  EF 000A1   EXTZV   #0, #2, 11(R1), -(SP)                     ;
                               6E    2E   A0  91 000A7       CMPB    46(R0), (SP)                              ;
                                          04  1B 000AB       BLEQU   4$                                        ;
                               6E    2E   A0  9A 000AD       MOVZBL  46(R0), (SP)                              ;
                                     24   BE  DD 000B1 4$:   PUSHL   @36(SP)                                   ; 2270
                                          56  DD 000B4       PUSHL   LOCAL_HEADER                              ;
                                          05  DD 000B6       PUSHL   #5                                        ;
                      0000G    CF         04  FB 000B8       CALLS   #4, CHECK_PROTECT                         ;
                         18    AE         01  D0 000BD       MOVL    #1, CTL_ACC_GRANTED                       ; 2277
                                  F78A CF49  7F 000C1 5$:    PUSHAQ  ATC+3[CODE]                               ; 2285
                               51         9E  90 000C6       MOVB    @(SP)+, ACTION                            ;
                                  F77F CF49  7F 000C9        PUSHAQ  ATC[CODE]                                 ; 2286
                    07         9E         00  E1 000CE       BBC     #0, @(SP)+, 6$                            ;
                               0F         51  91 000D2       CMPB    ACTION, #15                               ;
                                          02  13 000D5       BEQL    6$                                        ;
                                          51  94 000D7       CLRB    ACTION                                    ; 2287
                                  F770 CF49  7F 000D9 6$:    PUSHAQ  ATC+1[CODE]                               ; 2294
                    08         00         9E  8F 000DE       CASEB   @(SP)+, #0, #8                            ;
        0044      001E       0018       0012     000E2 7$:   .WORD   8$-7$,-                                   ;
        0012      0080       007A       0074     000EA               9$-7$,-                                   ;
                                        0012     000F2               10$-7$,-                                  ;
                                                                     12$-7$,-                                  ;
                                                                     14$-7$,-                                  ;
                                                                     15$-7$,-                                  ;
                                                                     16$-7$,-                                  ;
                                                                     8$-7$,-                                   ;
                                                                     6$-7$                                     ;
                               50    40   AE  9E 000F4 8$:   MOVAB   ATT_BUFFER, R0                            ;
                                          70  11 000F8       BRB     18$                                       ;
                               50    20   BE  D0 000FA 9$:   MOVL    @32(SP), R0                               ; 2299
                                          6A  11 000FE       BRB     18$                                       ;
                                  F74A CF49  7F 00100 10$:   PUSHAQ  ATC+2[CODE]                               ; 2302
```

```
                        52              9E  9A 00105         MOVZBL  @(SP)+, R2                              ;
                             F744  CF49  7F 00108         PUSHAQ  ATC+4[CODE]                             ;
                        50              9E  3C 0010D         MOVZWL  @(SP)+, R0                              ;
                        52              50  C0 00110         ADDL2   R0, R2                                  ;
                        50              66  9A 00113         MOVZBL  (LOCAL_HEADER), R0                      ; 2303
                        50              02  C4 00116         MULL2   #2, R0                                  ;
                        50              52  D1 00119         CMPL    R2, R0                                  ;
                                        03  1B 0011C         BLEQU   11$                                     ;
                        51              04  90 0011E         MOVB    #4, ACTION                              ; 2304
                        50              56  D0 00121 11$:    MOVL    LOCAL_HEADER, R0                        ; 2305
                                        44  11 00124         BRB     18$                                     ;
                             F724  CF49  7F 00126 12$:    PUSHAQ  ATC+2[CODE]                             ; 2309
                        53              9E  9A 0012B         MOVZBL  @(SP)+, R3                              ;
                             F71E  CF49  7F 0012E         PUSHAQ  ATC+4[CODE]                             ;
                        50              9E  3C 00133         MOVZWL  @(SP)+, R0                              ;
                        53              50  C0 00136         ADDL2   R0, R3                                  ;
                        52        01    A6  9A 00139         MOVZBL  1(LOCAL_HEADER), R2                     ; 2310
                        52              02  C4 0013D         MULL2   #2, R2                                  ;
                        50              66  9A 00140         MOVZBL  (LOCAL_HEADER), R0                      ; 2311
                        50              02  C4 00143         MULL2   #2, R0                                  ;
                        52              50  C2 00146         SUBL2   R0, R2                                  ;
                        52              53  D1 00149         CMPL    R3, R2                                  ;
                                        03  1B 0014C         BLEQU   13$                                     ;
                        51              04  90 0014E         MOVB    #4, ACTION                              ; 2312
                        50              66  9A 00151 13$:    MOVZBL  (LOCAL_HEADER), R0                      ; 2313
                                        10  11 00154         BRB     17$                                     ;
                        50        01    A6  9A 00156 14$:    MOVZBL  1(LOCAL_HEADER), R0                     ; 2315
                                        0A  11 0015A         BRB     17$                                     ;
                        50        02    A6  9A 0015C 15$:    MOVZBL  2(LOCAL_HEADER), R0                     ; 2316
                                        04  11 00160         BRB     17$                                     ;
                        50        03    A6  9A 00162 16$:    MOVZBL  3(LOCAL_HEADER), R0                     ; 2317
                        50            6640  3E 00166 17$:    MOVAW   (LOCAL_HEADER)[R0], R0                  ;
                             F6E0  CF49  7F 0016A 18$:    PUSHAQ  ATC+2[CODE]                             ; 2320
                        57              9E  9A 0016F         MOVZBL  @(SP)+, ADDRESS                         ;
                        57              50  C0 00172         ADDL2   R0, ADDRESS                             ;
              14        00              51  8F 00175         CASEB   ACTION, #0, #20                         ; 2326
  02E9        02E4      0235          02E9     00179 19$:    .WORD   64$-19$,-                               ;
  0162        014C      02E9          02E9     00181                 55$-19$,-                               ;
  01AA        00C7      002D          0169     00189                 63$-19$,-                               ;
  01E9        02E9      0086          01B9     00191                 64$-19$,-                               ;
  00E4        02E9      00AF          0230     00199                 64$-19$,-                               ;
                                      011F     001A1                 64$-19$,-                               ;
                                                                     39$-19$,-                               ;
                                                                     41$-19$,-                               ;
                                                                     42$-19$,-                               ;
                                                                     20$-19$,-                               ;
                                                                     29$-19$,-                               ;
                                                                     46$-19$,-                               ;
                                                                     47$-19$,-                               ;
                                                                     25$-19$,-                               ;
                                                                     64$-19$,-                               ;
                                                                     50$-19$,-                               ;
                                                                     54$-19$,-                               ;
                                                                     28$-19$,-                               ;
                                                                     64$-19$,-                               ;
                                                                     32$-19$,-                               ;
                                                                     36$-19$                                 ;
```

```
                                02BC  31 001A3          BRW     64$                                             ;
                          50      67  D0 001A6 20$:     MOVL    (ADDRESS), UIC                                  ; 2340
                          50   00 BE  9B 001A9          MOVZBW  @P, UIC                                         ; 2341
                          02      58  D1 001AD          CMPL    COUNT, #2                                       ; 2342
                                  0C  19 001B0          BLSS    21$                                             ;
                     52           6E  01 C1 001B2       ADDL3   #1, P, R2                                       ; 2343
                                  51  62 9A 001B6       MOVZBL  (R2), R1                                        ;
                50   10           10  51 F0 001B9       INSV    R1, #16, #16, UIC                               ;
                                  56  DD 001BE 21$:     PUSHL   LOCAL_HEADER                                    ; 2344
                              24  BE  DD 001C0          PUSHL   @36(SP)                                         ;
                                  50  DD 001C3          PUSHL   UIC                                             ;
                           0000V  CF      03 FB 001C5   CALLS   #3, CHANGE_OWNER                                ;
                              10  AE      50 D0 001CA   MOVL    R0, STATUS                                      ;
                                  4C  10 AE  E9 001CE   BLBC    STATUS, 26$                                     ; 2345
                                  56  04 AA  D0 001D2   MOVL    4(BASE), LOCAL_HEADER                           ; 2346
                                  03      58  D1 001D6   CMPL    COUNT, #3                                       ; 2347
                                          08  19 001D9   BLSS    22$                                             ;
                     50           6E      02  C1 001DB   ADDL3   #2, P, R0                                      ; 2348
                           40     A6      60  90 001DF   MOVB    (R0), 64(LOCAL_HEADER)                          ;
                                  04      58  D1 001E3 22$:  CMPL    COUNT, #4                                   ; 2349
                                          08  19 001E6   BLSS    23$                                             ;
                     50           6E      03  C1 001E8   ADDL3   #3, P, R0                                      ; 2350
                           41     A6      60  90 001EC   MOVB    (R0), 65(LOCAL_HEADER)                          ;
                                  05      58  D1 001F0 23$:  CMPL    COUNT, #5                                   ; 2351
                                          60  19 001F3   BLSS    31$                                             ;
                     50           6E      04  C1 001F5   ADDL3   #4, P, R0                                      ; 2352
                           34     A6      60  90 001F9 24$:  MOVB    (R0), 52(LOCAL_HEADER)                      ;
                                          26  11 001FD   BRB     27$                                             ; 2353
                              28  AE      67  D0 001FF 25$:  MOVL    (ADDRESS), UIC                              ; 2358
                        28    AE      00  BE      58  28 00203  MOVC3   COUNT, @P, UIC                          ; 2359
                                          56  DD 00209   PUSHL   LOCAL_HEADER                                    ; 2360
                                      24  BE  DD 0020B   PUSHL   @36(SP)                                         ;
                                      30  AE  DD 0020E   PUSHL   UIC                                             ;
                                   0000V  CF      03 FB 00211   CALLS   #3, CHANGE_OWNER                         ;
                                      10  AE      50 D0 00216   MOVL    R0, STATUS                              ;
                                          56  04 AA  D0 0021A   MOVL    4(BASE), LOCAL_HEADER                   ; 2361
                                          03  10 AE  E8 0021E 26$:  BLBS    STATUS, 27$                         ; 2362
                                                0248  31 00222   BRW     66$                                     ;
                                                0179  31 00225 27$:  BRW     52$                                 ; 2363
                14                00      00  BE      58  2C 00228 28$.  MOVC5   COUNT, @P, #0, #20, CLASS_BLOCK ; 2368
                                      2C  AE          0022E                                                       ;
                                      2C  AE  9F 00230      PUSHAB  CLASS_BLOCK                                   ; 2369
                                          56  DD 00233      PUSHL   LOCAL_HEADER                                  ;
                                   0000V  CF      02 FB 00235   CALLS   #2, CHANGE_CLASS                         ;
                                      10  AE      50 D0 0023A   MOVL    R0, STATUS                              ;
                                          DE  11 0023E      BRB     26$                                           ; 2370
                                      40  A6  00 BE  90 00240 29$:  MOVB    @P, 64(LOCAL_HEADER)                  ; 2375
                                          02  58 D1 00245      CMPL    COUNT, #2                                 ; 2376
                                              08 19 00248    BLSS    30$                                       ;
                                 50           6E  01 C1 0024A   ADDL3   #1, P, R0                               ; 2377
                                      41      A6  60 90 0024E   MOVB    (R0), 65(LOCAL_HEADER)                    ;
                                          03  58 D1 00252 30$:  CMPL    COUNT, #3                                 ; 2378
                                              CE 19 00255 31$:  BLSS    27$                                       ;
                                 50           6E  02 C1 00257   ADDL3   #2, P, R0                               ; 2379
                                              9C 11 0025B   BRB     24$                                         ;
                                          67  00 BE  90 0025D 32$:  MOVB    @P, (ADDRESS)                         ; 2385
                                          51  90 AA  D0 00261   MOVL    -112(BASE), R1                            ; 2386
```

```
                        50        10  AA  D0 00265          MOVL    16(BASE), R0                                    ;
              51    0B  A1          02  00  EF 00269          EXTZV   #0, #2, 11(R1), R1                              ;
                        51        2E  A0  91 0026F          CMPB    46(R0), R1                                      ;
                                      04  1B 00273          BLEQU   33$                                             ;
                        51        2E  A0  9A 00275          MOVZBL  46(R0), R1                                      ;
                    53  03            51  CB 00279 33$:     BICL3   R1, #3, MODE                                    ;
                                      50  D4 0027D          CLRL    J                                               ; 2387
                    51  50            01  78 0027F 34$:     ASHL    #1, J, R1                                       ; 2390
              52    67  02            51  EF 00283          EXTZV   R1, #2, (ADDRESS), R2                           ;
                        53            52  D1 00288          CMPL    R2, MODE                                        ;
                                      05  1B 0028B          BLEQU   35$                                             ;
              67        02            53  F0 0028D          INSV    MODE, R1, #2, (ADDRESS)                         ; 2391
                    E9  50            03  F3 00292 35$:     AOBLEQ  #3, J, 34$                                      ; 2387
                                      2B  11 00296          BRB     38$                                             ; 2326
              14        20  00    BE    58  2C 00298 36$:     MOVC5   COUNT, @P, #32, #20, (ADDRESS)                  ; 2397
                                                67  0029E
                        51        01  A6  9A 0029F          MOVZBL  1(LOCAL_HEADER), R1                             ; 2398
                        50        66  9A 002A3          MOVZBL  (LOCAL_HEADER), R0                              ;
                        51        50  C2 002A6          SUBL2   R0, R1                                          ;
                        3C        51  D1 002A9          CMPL    R1, #60                                         ; 2399
                                  73  1F 002AC          BLSSU   45$                                             ;
                        50        EC  A8  9E 002AE          MOVAB   -20(R8), R0                                     ; 2400
                                      02  18 002B2          BGEQ    37$                                             ;
                                      50  D4 002B4          CLRL    R0                                              ;
                    59  6E            14  C1 002B6 37$:     ADDL3   #20, P, R9                                      ; 2401
    0042    8F  20  69            50  2C 002BA          MOVC5   R0, (R9), #32, #66, 54(ADDRESS)                 ;
                                  36  A7  002C1                                                                     ;
                                      5C  11 002C3 38$:     BRB     45$                                             ; 2326
                    24    AE      46  AE  9E 002C5 39$:     MOVAB   ATT_BUFFER+6, T                                 ; 2405
                        51        C1  A6  9A 002CA 40$:     MOVZBL  1(LOCAL_HEADER), R1                             ; 2406
                        50        66  9A 002CE          MOVZBL  (LOCAL_HEADER), R0                              ;
                        51        50  C2 002D1          SUBL2   R0, R1                                          ;
                        3C        51  D1 002D4          CMPL    R1, #60                                         ; 2407
                                  1D  1E 002D7          BGEQU   43$                                             ;
                                      24  11 002D9          BRB     44$                                             ; 2409
                    24    AE      4C  AE  9E 002DB 41$:     MOVAB   ATT_BUFFER+12, T                                ; 2416
                                      E8  11 002E0          BRB     40$                                             ; 2417
                    24    AE      4E  AE  9E 002E2 42$:     MOVAB   ATT_BUFFER+14, T                                ; 2427
                        51        01  A6  9A 002E7          MOVZBL  1(LOCAL_HEADER), R1                             ; 2428
                        50        66  9A 002EB          MOVZBL  (LOCAL_HEADER), R0                              ;
                        51        50  C2 002EE          SUBL2   R0, R1                                          ;
                        3C        51  D1 002F1          CMPL    R1, #60                                         ; 2429
                                  09  1F 002F4          BLSSU   44$                                             ;
    0042    8F  20              6E    00  2C 002F6 43$:     MOVC5   #0, (SP), #32, #66, 54(ADDRESS)                 ; 2430
                                  36  A7  002FD                                                                     ;
                                  40  AE  9F 002FF 44$:     PUSHAB  ATT_BUFFER                                      ; 2431
                                      57  DD 00302          PUSHL   ADDRESS                                         ;
                                      14  DD 00304          PUSHL   #20                                             ;
                    0000G CF          03  FB 00306          CALLS   #3, MAKE_NAMEBLOCK                              ;
              24    BE    00    BE    58  28 0030B          MOVC3   COUNT, @P, @T                                   ; 2432
              14    20          6E    00  2C 00311          MOVC5   #0, (SP), #32, #20, (ADDRESS)                   ; 2433
                                                67  00316                                                            ;
                                      57  DD 00317          PUSHL   ADDRESS                                         ; 2434
                                  44  AE  9F 00319          PUSHAB  ATT_BUFFER                                      ;
                    0000G CF          02  FB 0031C          CALLS   #2, MAKE_STRING                                 ;
                                      3D  11 00321 45$:     BRB     49$                                             ; 2326
              0D    30    00    BE    58  2C 00323 46$:     MOVC5   COUNT, @P, #48, #13, ATT_BUFFER                 ; 2438
```

```
                                40  AE      00329
                                57  DD 0032B          PUSHL   ADDRESS                                                   ; 2439
                                44  AE  9F 0032D      PUSHAB  ATT_BUFFER                                              ;
                                29  11 00330          BRB     48$                                                     ;
        2C      30      00  BE      58  2C 00332 47$: MOVC5   COUNT, @P, #48, #44, ATT_BUFFER                        ; 2443
                                40  AE      00338                                                                     ;
                14  A7      40  AE  B0 0033A          MOVW    ATT_BUFFER, 20(ADDRESS)                                 ; 2444
                                1E  A7  9F 0033F      PUSHAB  30(ADDRESS)                                             ; 2445
                                46  AE  9F 00342      PUSHAR  ATT_BUFFER+2                                            ;
          0000V CF              02  FB 00345          CALLS   #2, CONVERT_DATE                                        ;
                                16  A7  9F 0034A      PUSHAB  22(ADDRESS)                                             ; 2446
                                53  AE  9F 0034D      PUSHAB  ATT_BUFFER+15                                           ;
          0000V CF              02  FB 00350          CALLS   #2, CONVERT_DATE                                        ;
                                26  A7  9F 00355      PUSHAB  38(ADDRESS)                                             ; 2447
                                60  AE  9F 00358      PUSHAB  ATT_BUFFER+28                                           ;
          0000V CF              02  FB 0035B 48$:     CALLS   #2, CONVERT_DATE                                        ;
                                44  11 00360 49$:     BRB     53$                                                     ; 2326
                50      10  AA  D0 00362 50$:     MOVL    16(BASE), R0                                            ; 2453
                33      34  A0  E9 00366          BLBC    52(R0), 51$                                             ;
                                56  DD 0036A          PUSHL   LOCAL_HEADER                                            ; 2456
          0000G CF              01  FB 0036C          CALLS   #1, CHECKSUM                                            ;
                                56  DD 00371          PUSHL   LOCAL_HEADER                                            ; 245
          0000G CF              01  FB 00373          CALLS   #1, MARK_DIRTY                                          ;
            14  AE              01  D0 00378          MOVL    #1, ACL_MODIFIED                                        ; 2458
                                6E  DD 0037C          PUSHL   P                                                       ; 2460
                7E              57  7D 0037E          MOVQ    ADDRESS, -(SP)                                          ;
                                59  DD 00381          PUSHL   CODE                                                    ;
          0000G CF              04  FB 00383          CALLS   #4, ACL_DISPATCH                                        ;
            10  AE              50  D0 00388          MOVL    R0, STATUS                                              ;
                56      04  AA  D0 0038C          MOVL    4(BASE), LOCAL_HEADER                                   ; 2461
                09      10  AE  E8 00390          BLBS    STATUS, 1$                                              ; 2462
        50      1C  AE          34  C1 00394          ADDL3   #52, 28(SP), R0                                         ; 2463
                60      10  AE  D0 00399          MOVL    STATUS, (R0)                                            ;
                10  AE          01  D0 0039D 51$:     MOVL    #1, STATUS                                              ; 2465
                01  AA      40  8F  88 003A1 52$:     BISB2   #64, 1(BASE)                                            ; 2466
                    00B9        .   003A6 53$:     BRW     64$                                                     ; 2326
                02              58  D1 003A9 54$:     CMPL    COUNT, #2                                               ; 2474
                                06  18 003AC          BGEQ    56$                                                     ;
                10  AE          34  D0 003AE 55$:     MOVL    #52, STATUS                                             ;
                                6B  11 003B2          BRB     60$                                                     ;
                50      01  A6  9A 003B4 56$:     MOVZBL  1(LOCAL_HEADER), R0                                     ; 2479
                51          6640  3E 003B8          MOVAW   (LOCAL_HEADER)[R0], R1                                  ;
                50      3A  A6  9A 003BC          MOVZBL  58(LOCAL_HEADER), R0                                    ; 2480
                51          6140  3E 003C0          MOVAW   (R1)[R0], MAP_END                                       ; 2479
                50      02  A6  9A 003C4          MOVZBL  2(LOCAL_HEADER), R0                                     ; 2481
                50          6640  3E 003C8          MOVAW   (LOCAL_HEADER)[R0], ACL_END                             ;
                53      03  A6  9E 003CC          MOVAB   3(LOCAL_HEADER), R3                                     ; 2482
                52              63  9A 003D0 57$:     MOVZBL  (R3), R2                                                ;
                52          6642  3E 003D3          MOVAW   (LOCAL_HEADER)[R2], R2                                  ;
                52              50  D1 003D7          CMPL    ACL_END, R2                                             ;
                                0C  18 003DA          BGEQ    58$                                                     ;
                                60  95 003DC          TSTB    (ACL_END)                                               ; 2485
                                08  13 003DE          BEQL    58$                                                     ;
                52              60  9A 003E0          MOVZBL  (ACL_END), R2                                           ; 2486
                50              52  C0 003E3          ADDL2   R2, ACL_END                                             ;
                                E8  11 003E6          BRB     57$                                                     ; 2482
                63      02  A6  91 003E8 58$:     CMPB    2(LOCAL_HEADER), (R3)                                   ; 2490
```

```
                                    03 12 003EC          BNEQ    59$                                         ;
                    50              51 D0 003EE          MOVL    MAP_END, R0                                 ; 2491
            50      56              50 C3 003F1 59$:     SUBL3   R0, LOCAL_HEADER, R0                        ; 2489
                    59        01FE  C0 9E 003F5          MOVAB   510(R0), RESERVED_LENGTH                    ; 2488
            51 000001FE 8F          59 C3 003FA          SUBL3   RESERVED_LENGTH, #510, R1                   ; 2493
            50      56              59 C3 00402          SUBL3   RESERVED_LENGTH, LOCAL_HEADER, R0           ; 2495
   51       00      6E              00 2C 00406          MOVC5   #0, (SP), #0, R1, 510(R0)                   ;
                              01FE  C0    0040B                                                              ;
                    6E              02 C0 0040E          ADDL2   #2, P                                       ; 2496
                    58              02 C2 00411          SUBL2   #2, COUNT                                   ; 2497
                    59              58 D1 00414          CMPL    COUNT, RESERVED_LENGTH                      ; 2498
                                    08 15 00417          BLEQ    61$                                         ;
              10    AE        08C8  8F 3C 00419          MOVZWL  #2248, STATUS                               ;
                                    4C 11 0041F 60$:     BRB     66$                                         ;
            50      56              58 C3 00421 61$:     SUBL3   COUNT, LOCAL_HEADER, R0                     ; 2499
                    50        01FE  C0 9E 00425          MOVAB   510(R0), R0                                 ;
            57      50              01 CB 0042A          BICL3   #1, R0, ADDRESS                             ;
            50      57              56 C3 0042E          SUBL3   LOCAL_HEADER, ADDRESS, R0                   ; 2503
                    50              02 C6 00432          DIVL2   #2, R0                                      ;
              03    A6          02  A6 91 00435          CMPB    2(LOCAL_HEADER), 3(LOCAL_HEADER)            ; 2500
                                    1D 12 0043A          BNEQ    62$                                         ;
                    51          01  A6 9A 0043C          MOVZBL  1(LOCAL_HEADER), R1                         ; 2501
                    52          3A  A6 9A 00440          MOVZBL  58(LOCAL_HEADER), R2                        ;
                    51              52 C0 00444          ADDL2   R2, R1                                      ;
   51       02      A6              08 00 ED 00447       CMPZV   #0, #8, 2(LOCAL_HEADER), R1                 ;
                                    0A 13 0044D          BEQL    62$                                         ;
              03    A6              50 90 0044F          MOVB    R0, 3(LOCAL_HEADER)                         ; 2503
              02    A6              50 90 00453          MOVB    R0, 2(LOCAL_HEADER)                         ;
                                    04 11 00457          BRB     63$                                         ; 2502
              03    A6              50 90 00459 62$:     MOVB    R0, 3(LOCAL_HEADER)                         ; 2504
            67      00      BE      58 28 0045D 63$:     MOVC3   COUNT, @P, (ADDRESS)                        ; 2505
            02      04      AE  08  AE F2 00462 64$:     AOBLSS  8(SP), I, 65$                               ; 2222
                                    03 11 00468          BRB     66$                                         ;
                                  FBE0 31 0046A 65$:     BRW     1$                                          ;
            2B              5B      0D E1 0046D 66$:     BBC     #13, SAVE_CHAR, 67$                         ; 2516
            26        35    A6      05 E0 00471          BBS     #5, 53(LOCAL_HEADER), 67$                   ; 2517
                   0000G    CF      00 FB 00476          CALLS   #0, UPDATE_DIRSEQ                           ; 2520
                    50          18  AA D0 0047B          MOVL    24(BASE), R0                                ; 2521
                          0080 CA40 DD 0047F             PUSHL   128(BASE)[R0]                               ;
                                    01 DD 00484          PUSHL   #1                                          ;
                   0000G    CF      02 FB 00486          CALLS   #2, KILL_BUFFERS                            ;
                    50          20  BE D0 0048B          MOVL    @32(SP), R0                                 ; 2523
                              00B0  C0 D5 0048F          TSTL    176(R0)                                     ;
                                    07 13 00493          BEQL    67$                                         ;
                                    50 DD 00495          PUSHL   R0                                          ; 2525
                   0000G    CF      01 FB 00497          CALLS   #1, KILL_DINDX                              ;
                    50          04  AC D0 0049C 67$:     MOVL    HEADER, R0                                  ; 2532
            50        34    A0 0001D080 8F CB 004A0      BICL3   #118912, 52(R0), R0                         ;
                    5B FFFE2F7F 8F CA 004A9              BICL2   #-118913, R11                               ; 2533
         34 A6      5B              50 C9 004B0          BISL3   R0, R11, 52(LOCAL_HEADER)                   ;
              18    A6          0C  AE D0 004B5          MOVL    SAVE_HIBLK, 24(LOCAL_HEADER)                ; 2534
              0A    AE          14  AE E9 004BA          BLBC    ACL_MODIFIED, 68$                           ; 2539
                                20  BE DD 004BE          PUSHL   @32(SP)                                     ; 2541
                   0000G    CF      01 FB 004C1          CALLS   #1, ACL_BUILDACL                            ;
                                    0E 11 004C6          BRB     69$                                         ;
                                    56 DD 004C8 68$:     PUSHL   LOCAL_HEADER                                ; 2544
                   0000G    CF      01 FB 004CA          CALLS   #1, CHECKSUM                                ;
```

```
                          56  DD 004CF          PUSHL   LOCAL_HEADER                          ; 2545
                          01  FB 004D1          CALLS   #1, MARK_DIRTY                      ;
                          00  FB 004D6 69$:     CALLS   #0, PMS_END_SUB                     ; 2551
                      10  AE  E8 004DB          BLBS    STATUS, 70$                         ; 2553
                      10  AE  BF 004DF          CHMU    STATUS                              ;
                          04 004E2 70$:         RET                                         ; 2555

; Routine Size:  1251 bytes,    Routine Base:  $CODE$ + 07B3
```

```
;  1572          2556  1 ROUTINE CONVERT_DATE (STRING, TIME_BLOCK) : L_NORM NOVALUE =
;  1573          2557  1
;  1574          2558  1 !++
;  1575          2559  1 !
;  1576          2560  1 ! FUNCTIONAL DESCRIPTION:
;  1577          2561  1 !
;  1578          2562  1 !       This routine converts a files-11 structure level 1 ASCII date/time
;  1579          2563  1 !       string into 64 bit binary format.
;  1580          2564  1 !
;  1581          2565  1 !
;  1582          2566  1 ! CALLING SEQUENCE:
;  1583          2567  1 !       CONVERT_DATE (ARG1, ARG2)
;  1584          2568  1 !
;  1585          2569  1 ! INPUT PARAMETERS:
;  1586          2570  1 !       ARG1: address of date/time string
;  1587          2571  1 !
;  1588          2572  1 ! IMPLICIT INPUTS:
;  1589          2573  1 !       NONE
;  1590          2574  1 !
;  1591          2575  1 ! OUTPUT PARAMETERS:
;  1592          2576  1 !       ARG2: address of quadword buffer
;  1593          2577  1 !
;  1594          2578  1 ! IMPLICIT OUTPUTS:
;  1595          2579  1 !       NONE
;  1596          2580  1 !
;  1597          2581  1 ! ROUTINE VALUE:
;  1598          2582  1 !       NONE
;  1599          2583  1 !
;  1600          2584  1 ! SIDE EFFECTS:
;  1601          2585  1 !       NONE
;  1602          2586  1 !
;  1603          2587  1 !--
;  1604          2588  1
;  1605          2589  2 BEGIN
;  1606          2590  2
;  1607          2591  2 LITERAL
;  1608          2592  2         DATLEN          = 20;           ! length of date/time string
;  1609          2593  2
;  1610          2594  2 LOCAL
;  1611          2595  2         DATDESC         : VECTOR [2],   ! string descriptor for date string
;  1612          2596  2         DATBUF          : VECTOR [DATLEN, BYTE];
;  1613          2597  2                                         ! buffer to build expanded string
;  1614          2598  2
;  1615          2599  2
;  1616          2600  2 ! Copy the given string into the buffer, inserting the date punctuation
;  1617          2601  2 ! as appropriate. Then convert with the system service.
;  1618          2602  2 !
;  1619          2603  2
;  1620          2604  2 (DATBUF+00)<0,16> = .(.STRING);
;  1621          2605  2 (DATBUF+02)<0,8>  = '-';
;  1622          2606  2 (DATBUF+03)<0,24> = .(.STRING+2);
;  1623          2607  2 (DATBUF+06)<0,24> = '-19';
;  1624          2608  2 (DATBUF+09)<0,16> = .(.STRING+5);
;  1625          2609  2 (DATBUF+11)<0,8>  = ' ';
;  1626          2610  2 (DATBUF+12)<0,16> = .(.STRING+7);
;  1627          2611  2 (DATBUF+14)<0,8>  = ':';
;  1628          2612  2 (DATBUF+15)<0,16> = .(.STRING+9);
```

```
;  1629          2613  2 (DATBUF+17)<0,8>  = ':';
;  1630          2614  2 (DATBUF+18)<0,16> = .(.STRING+11);
;  1631          2615  2 DATDESC[0] = DATLEN;
;  1632          2616  2 DATDESC[1] = DATBUF;
;  1633          2617  2 $BINTIM (TIMBUF = DATDESC, TIMADR = .TIME_BLOCK);
;  1634          2618  2
;  1635          2619  1 END;                                    ! end of routine CONVERT_DATE


                                                                .EXTRN  SYS$BINTIM

                                                        0000 00000 CONVERT_DATE:
                                                                .WORD   Save nothing                                    ; 2556
                                                5E              1C  C2 00002    SUBL2   #28, SP                         ;
                                                50      04      AC  D0 00005    MOVL    STRING, R0                      ; 2604
                                                6E              60  B0 00009    MOVW    (R0), DATBUF                    ;
                                            02  AE              2D  90 0000C    MOVB    #45, DATBUF+2                   ; 2605
      03    AE          18                  00      02      A0  F0 00010    INSV    2(R0), #0, #24, DATBUF+3        ; 2606
      06    AE          18                  00 0039312D     8F  F0 00017    INSV    #3748141, #0, #24, DATBUF+6     ; 2607
                                            09  AE      05      A0  B0 00021    MOVW    5(R0), DATBUF+9                 ; 2608
                                            0B  AE              20  90 00026    MOVB    #32, DATBUF+11                  ; 2609
                                            0C  AE      07      A0  B0 0002A    MOVW    7(R0), DATBUF+12                ; 2610
                                            0E  AE              3A  90 0002F    MOVB    #58, DATBUF+14                  ; 2611
                                            0F  AE      09      A0  B0 00033    MOVW    9(R0), DATBUF+15                ; 2612
                                            11  AE              3A  90 00038    MOVB    #58, DATBUF+17                  ; 2613
                                            12  AE      0P      A0  B0 0003C    MOVW    11(R0), DATBUF+18               ; 2614
                                            14  AE              14  D0 00041    MOVL    #20, DATDESC                    ; 2615
                                            18  AE              6E  9E 00045    MOVAB   DATBUF, DATDESC+4               ; 2616
                                                        08      AC  DD 00049    PUSHL   TIME_BLOCK                      ; 2617
                                                        18      AE  9F 0004C    PUSHAB  DATDESC                         ;
                                    00000000G   00              02  FB 0004F    CALLS   #2, SYS$BINTIM                  ;
                                                                04  00056      RET                                     ; 2619

; Routine Size:  87 bytes,    Routine Base:  $CODE$ + 0C96
```

```
;1637     2620  1 GLOBAL ROUTINE CHANGE_OWNER (UIC, ORG_FCB, ORG_HEADER) : L_NORM =
;1638     2621  1
;1639     2622  1 !++
;1640     2623  1 !
;1641     2624  1 ! FUNCTIONAL DESCRIPTION:
;1642     2625  1 !
;1643     2626  1 !       This routine changes the owner UIC of a file. It check for privilege
;1644     2627  1 !       and then chains through all the headers of the file, changing the
;1645     2628  1 !       owner UIC, crediting the blocks to the old owner and charging them
;1646     2629  1 !       to the new owner, if necessary.
;1647     2630  1 !
;1648     2631  1 ! CALLING SEQUENCE:
;1649     2632  1 !       CHANGE_OWNER (ARG1, ARG2, ARG3)
;1650     2633  1 !
;1651     2634  1 ! INPUT PARAMETERS:
;1652     2635  1 !       ARG1: new UIC
;1653     2636  1 !       ARG2: address of file's FCB
;1654     2637  1 !       ARG3: address of file header or 0 if it must be read
;1655     2638  1 !
;1656     2639  1 ! IMPLICIT INPUTS:
;1657     2640  1 !       CLEANUP_FLAGS: cleanup action and status flags
;1658     2641  1 !
;1659     2642  1 ! OUTPUT PARAMETERS:
;1660     2643  1 !       NONE
;1661     2644  1 !
;1662     2645  1 ! IMPLICIT OUTPUTS:
;1663     2646  1 !       NONE
;1664     2647  1 !
;1665     2648  1 ! ROUTINE VALUE:
;1666     2649  1 !       NONE
;1667     2650  1 !
;1668     2651  1 ! SIDE EFFECTS:
;1669     2652  1 !       file headers read and written, quota file entries modified, FCB modified
;1670     2653  1 !
;1671     2654  1 !--
;1672     2655  1
;1673     2656  2 BEGIN
;1674     2657  2
;1675     2658  2 LINKAGE
;1676     2659  2         L_SEARCH_RIGHT  = JSB (REGISTER = 2, REGISTER = 4;
;1677     2660  2                               REGISTER = 1, REGISTER = 5);
;1678     2661  2
;1679     2662  2 MAP
;1680     2663  2         ORG_FCB         : REF BBLOCK;   ! FCB of file
;1681     2664  2
;1682     2665  2 LOCAL
;1683     2666  2         ORG_FILE_OWNER,                 ! Original file owner
;1684     2667  2         HEADER          : REF BBLOCK,   ! pointer to current file header
;1685     2668  2         FCB             : REF BBLOCK,   ! FCB of current header, if any
;1686     2669  2         ARB             : REF BBLOCK,   ! Access rights block address
;1687     2670  2         RIGHTS_DESC     : REF BBLOCK,   ! Rights list descr address
;1688     2671  2         ID_FOUND        : REF BBLOCK,   ! Addr of matching ID
;1689     2672  2         RIGHTS_SEG      : REF BBLOCK,   ! Addr of rights segment that has ID
;1690     2673  2         STATUS_1,                       ! Indicates file owner resource
;1691     2674  2         STATUS_2,                       ! Indicates target UIC resource
;1692     2675  2         SIZE;                           ! size of file section
;1693     2676  2
```

```
;  1694          2677  2 BIND_COMMON;
;  1695          2678  2
;  1696          2679  2 EXTERNAL ROUTINE
;  1697          2680  2         EXE$SEARCH_RIGHT        : L_SEARCH_RIGHT ADDRESSING_MODE (GENERAL),
;  1698          2681  2                                   ! Search rights list for identifier
;  1699          2682  2         CHARGE_QUOTA    : L_NORM,  ! charge blocks to user's quota
;  1700          2683  2         CHECKSUM        : L_NORM,  ! compute file header checksum
;  1701          2684  2         MARK_DIRTY      : L_NORM,  ! mark buffer for write-back
;  1702          2685  2         NEXT_HEADER     : L_NORM,  ! read next extension header
;  1703          2686  2         READ_HEADER     : L_NORM;  ! read a file header
;  1704          2687  2
;  1705          2688  2
;  1706          2689  2 ! Set up local pointers. Then check privilege. If the new UIC is the same
;  1707          2690  2 ! as the old UIC, this whole thing is a NOP (and does not require privilege).
;  1708          2691  2 !
;  1709          2692  2
;  1710          2693  2 FCB = .ORG_FCB;
;  1711          2694  2 ORG_FILE_OWNER = .FCB[FCB$L_FILEOWNER];
;  1712          2695  2
;  1713          2696  2 IF .ORG_FILE_OWNER EQL .UIC THEN RETURN 1;
;  1714          2697  2
;  1715          2698  2 ! If the UIC is not the same, it is necessary to determine whether or not
;  1716          2699  2 ! the requesting process is able to change the owner of the file.  In order
;  1717          2700  2 ! for the the owner to be changed, the requesting process must be: 1) running
;  1718          2701  2 ! with SYSPRV, 2) be running with GRPPRV and the group of the file owner,
;  1719          2702  2 ! the requesting process, and the target UIC must match, and 3) have resource
;  1720          2703  2 ! rights to the file owner's UIC and the target UIC.
;  1721          2704  2
;  1722          2705  2 ARB = .IO_PACKET[IRP$L_ARB];
;  1723          2706  2 RIGHTS_DESC = ARB[ARB$L_RIGHTSLIST];
;  1724          2707  2
;  1725          2708  2 ! Determine whether or not the accessor has resource rights to the file's
;  1726          2709  2 ! owner UIC.
;  1727          2710  2
;  1728          2711  2 STATUS_1 = 1;                              ! Assume resource rights to owner UIC
;  1729          2712  2 IF .ARB[ARB$L_UIC] NEQ .ORG_FILE_OWNER
;  1730          2713  2 THEN IF EXE$SEARCH_RIGHT (.ORG_FILE_OWNER, .RIGHTS_DESC;
;  1731          2714  2                          ID_FOUND, RIGHTS_SEG)
;  1732          2715  3      THEN (IF NOT .ID_FOUND[KGB$V_RESOURCE]
;  1733          2716  3            THEN STATUS_1 = 0)
;  1734          2717  2      ELSE STATUS_1 = 0;
;  1735          2718  2
;  1736          2719  2 ! Determine whether or not the accessor has resource rights to the target UIC.
;  1737          2720  2
;  1738          2721  2 STATUS_2 = 1;                              ! Assume resource rights to target UIC
;  1739          2722  2 IF .ARB[ARB$L_UIC] NEQ .UIC
;  1740          2723  2 THEN IF EXE$SEARCH_RIGHT (.UIC, .RIGHTS_DESC; ID_FOUND, RIGHTS_SEG)
;  1741          2724  3      THEN (IF NOT .ID_FOUND[KGB$V_RESOURCE]
;  1742          2725  3            THEN STATUS_2 = 0)
;  1743          2726  2      ELSE STATUS_2 = 0;
;  1744          2727  2
;  1745          2728  2 ! Now, check the results of the above tests.  If either fails, a check is
;  1746          2729  2 ! made to see if SYSPRV or GRPPRV apply.
;  1747          2730  2
;  1748          2731  2 $ASSUME ($BITPOSITION (UIC$V_GROUP), EQL, 16)
;  1749          2732  2 $ASSUME ($FIELDWIDTH (UIC$V_GROUP), EQL, 14)
;  1750          2733  2 $ASSUME ($BITPOSITION (UIC$V_FORMAT), EQL, 30)
```

```
;  1751          2734  2 $ASSUME ($FIELDWIDTH (UIC$V_FORMAT), EQL, 2)
;  1752          2735  2
;  1753          2736  2 IF NOT .STATUS_1 OR NOT .STATUS_2
;  1754          2737  2 THEN
;  1755          2738  3     BEGIN
;  1756          2739  3     IF NOT .BBLOCK[LOCAL_ARB[ARB$Q_PRIV], PRV$V_SYSPRV]
;  1757          2740  3     AND NOT .BBLOCK[LOCAL_ARB[ARB$Q_PRIV], PRV$V_BYPASS]
;  1758          2741  4     AND NOT (.BBLOCK[LOCAL_ARB[ARB$Q_PRIV], PRV$V_GRPPRV] AND
;  1759          2742  5              (.(ARB[ARB$L_UIC])<16,16> EQL .FCB[FCB$W_UICGROUP] AND
;  1760          2743  5               .(ARB[ARB$L_UIC])<16,16> EQL .UIC<16,16> AND
;  1761          2744  4               .BBLOCK [ARB[ARB$L_UIC], UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT))
;  1762          2745  3     THEN RETURN SS$_NOPRIV;
;  1763          2746  2     END;
;  1764          2747  2
;  1765          2748  2 ! Credit the original owner and charge the new owner for the blocks allocated
;  1766          2749  2 ! to the file. Loop through the FCB chain to count up the number of headers.
;  1767          2750  2
;  1768          2751  2 SIZE = .FCB[FCB$L_FILESIZE];
;  1769          2752  2 IF NOT .CLEANUP_FLAGS[CLF_HDRNOTCHG]
;  1770          2753  2 THEN
;  1771          2754  3     BEGIN
;  1772          2755  3     DO
;  1773          2756  4         BEGIN
;  1774          2757  4         SIZE = .SIZE + 1;
;  1775          2758  4         FCB = .FCB[FCB$L_EXFCB];
;  1776          2759  4         END
;  1777          2760  3     UNTIL .FCB EQL 0;
;  1778          2761  3     FCB = .ORG_FCB;
;  1779          2762  2     END;
;  1780          2763  2
;  1781          2764  2 IF .SIZE GTRU 0
;  1782          2765  2 THEN
;  1783          2766  3     BEGIN
;  1784          2767  3     CHARGE_QUOTA (.UIC, .SIZE, BITLIST (QUOTA_CHECK, QUOTA_CHARGE));
;  1785          2768  3     CHARGE_QUOTA (.ORG_FILE_OWNER, -.SIZE, BITLIST (QUOTA_CHARGE));
;  1786          2769  2     END;
;  1787          2770  2
;  1788          2771  2 ! Now loop, changing the owner UIC of a header and writing it. Note that no
;  1789          2772  2 ! provision is made for error recovery. This is because the previous ownerships
;  1790          2773  2 ! of the headers of a multi-header file could be different, and therefore
;  1791          2774  2 ! cannot be saved in finite space. Thus a failure here could leave the
;  1792          2775  2 ! file half changed.
;  1793          2776  2
;  1794          2777  2 HEADER = .ORG_HEADER;
;  1795          2778  2 IF .HEADER EQL 0
;  1796          2779  2 THEN HEADER = READ_HEADER (0, .FCB);
;  1797          2780  2
;  1798          2781  2 DO
;  1799          2782  3     BEGIN
;  1800          2783  3     HEADER[FH2$L_FILEOWNER] = .UIC;
;  1801          2784  3     CHECKSUM (.HEADER);
;  1802          2785  3     MARK_DIRTY (.HEADER);
;  1803          2786  3     HEADER = NEXT_HEADER (.HEADER, .FCB);
;  1804          2787  3     FCB = .FCB[FCB$L_EXFCB];
;  1805          2788  3     END
;  1806          2789  2 UNTIL .HEADER EQL 0;
;  1807          2790  2
```

```
2 ! If we chained to extension headers, reread the primary for further use.
2 !
2
2 IF .FCB NEQ .ORG_FCB
2 THEN
3     BEGIN
3     HEADER = READ_HEADER (0, .ORG_FCB);
2     END;
2
2 ! Mark the new owner in the FCB.
2
2 ORG_FCB[FCB$L_FILEOWNER] = .UIC;
2
2 RETURN 1;
2
1 END;                                          ! end of routine CHANGE_OWNER


                                                         .EXTRN   EXE$SEARCH_RIGHT
                                                         .EXTRN   CHARGE_QUOTA, NEXT_HEADER

                                       0BFC 00000        .ENTRY   CHANGE_OWNER, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 2620
                                                                  R11                                        ;
                58      0284      CA  9E 00002           MOVAB    644(BASE), R8                              ; 2675
                56        08      AC  D0 00007           MOVL     ORG_FCB, FCB                               ; 2693
                5B        58      A6  D0 0000B           MOVL     88(FCB), ORG_FILE_OWNER                    ; 2694
           04   AC                5B  D1 0000F           CMPL     ORG_FILE_OWNER, UIC                        ; 2696
                                  03  12 00013           BNEQ     1$                                         ;
                                00F5  31 00015           BRW      14$                                        ;
                50        90      AA  D0 00018 1$:       MOVL     -112(BASE), R0                             ; 2705
                53        58      A0  D0 0001C           MOVL     88(R0), ARB                                ;
                54        20      A3  9E 00020           MOVAB    32(R3), RIGHTS_DESC                        ; 2706
                59                01  D0 00024           MOVL     #1, STATUS_1                               ; 2711
                5B        38      A3  D1 00027           CMPL     56(ARB), ORG_FILE_OWNER                    ; 2712
                                  12  13 0002B           BEQL     3$                                         ;
                52                5B  D0 0002D           MOVL     ORG_FILE_OWNER, R2                         ; 2713
                     00000000G    00  16 00030           JSB      EXE$SEARCH_RIGHT                           ;
                04                50  E9 00036           BLBC     R0, 2$                                     ;
                02        04      A1  E8 00039           BLBS     4(ID_FOUND), 3$                            ; 2715
                                  59  D4 0003D 2$:       CLRL     STATUS_1                                   ; 2717
                57                01  D0 0003F 3$:       MOVL     #1, STATUS_2                               ; 2721
           04   AC        38      A3  D1 00042           CMPL     56(ARB), UIC                               ; 2722
                                  13  13 00047           BEQL     5$                                         ;
                52        04      AC  D0 00049           MOVL     UIC, R2                                    ; 2723
                     00000000G    00  16 0004D           JSB      EXE$SEARCH_RIGHT                           ;
                04                50  E9 00053           BLBC     R0, 4$                                     ;
                02        04      A1  E8 00056           BLBS     4(ID_FOUND), 5$                            ; 2724
                                  57  D4 0005A 4$:       CLRL     STATUS_2                                   ; 2726
                03                59  E9 0005C 5$:       BLBC     STATUS_1, 6$                               ; 2736
                26                57  E8 0005F           BLBS     STATUS_2, 8$                               ;
      22        68                1C  E0 00062 6$:       BBS      #28, (R8), 8$                              ; 2739
      1E        68                1D  E0 00066           BBS      #29, (R8), 8$                              ; 2740
      15   04   A8                02  E1 0006A           BBC      #2, 4(R8), 7$                              ; 2741
           5A   A6        3A      A3  B1 0006F           CMPW     58(ARB), 90(FCB)                           ; 2742
                                  0E  12 00074           BNEQ     7$                                         ;
           06   AC        3A      A3  B1 00076           CMPW     58(ARB), UIC+2                             ; 2743
```

```
                              07  12 0007B      BNEQ    7$                                      ;
         C0   8F        3B    A3  93 0007D      BITB    59(ARB), #192                           ; 2744
                              04  13 00082      BEQL    8$                                      ;
              50              24  D0 00084 7$:  MOVL    #36, R0                                 ; 2745
                                  04 00087      RET                                             ;
              52        38    A6  D0 00088 8$:  MOVL    56(FCB), SIZE                           ; 2751
   0C         6A              1B  E0 0008C      BBS     #27, (BASE), 10$                        ; 2752
                              52  D6 00090 9$:  INCL    SIZE                                    ; 2757
              56        0C    A6  D0 00092      MOVL    12(FCB), FCB                            ; 2758
                              F8  12 00096      BNEQ    9$                                      ; 2760
              56        08    AC  D0 00098      MOVL    ORG_FCB, FCB                            ; 2761
                              52  D5 0009C 10$: TSTL    SIZE                                    ; 2764
                              18  13 0009E      BEQL    11$                                     ;
                              03  DD 000A0      PUSHL   #3                                      ; 2767
                              52  DD 000A2      PUSHL   SIZE                                    ;
                        04    AC  DD 000A4      PUSHL   UIC                                     ;
       0000G  CF              03  FB 000A7      CALLS   #3, CHARGE_QUOTA                        ;
                              02  DD 000AC      PUSHL   #2                                      ; 2768
              7E              52  CE 000AE      MNEGL   SIZE, -(SP)                             ;
                              58  DD 000B1      PUSHL   ORG_FILE_OWNER                          ;
       0000G  CF              03  FB 000B3      CALLS   #3, CHARGE_QUOTA                        ;
              52        0C    AC  D0 000B8 11$: MOVL    ORG_HEADER, HEADER                      ; 2777
                              0C  12 000BC      BNEQ    12$                                     ; 2778
                              56  DD 000BE      PUSHL   FCB                                     ; 2779
                              7E  D4 000C0      CLRL    -(SP)                                   ;
       0000G  CF              02  FB 000C2      CALLS   #2, READ_HEADER                         ;
              52              50  D0 000C7      MOVL    R0, HEADER                              ;
         3C   A2        04    AC  D0 000CA 12$: MOVL    UIC, 60(HEADER)                         ; 2783
                              52  DD 000CF      PUSHL   HEADER                                  ; 2784
       0000G  CF              01  FB 000D1      CALLS   #1, CHECKSUM                            ;
                              52  DD 000D6      PUSHL   HEADER                                  ; 2785
       0000G  CF              01  FB 000D8      CALLS   #1, MARK_DIRTY                          ;
                      0044    8F  BB 000DD      PUSHR   #^M<R2,R6>                              ; 2786
       0000G  CF              02  FB 000E1      CALLS   #2, NEXT_HEADER                         ;
              52              50  D0 000E6      MOVL    R0, HEADER                              ;
              56        0C    A6  D0 000E9      MOVL    12(FCB), FCB                            ; 2787
                              52  D5 000ED      TSTL    HEADER                                  ; 2789
                              D9  12 000EF      BNEQ    12$                                     ;
         08   AC              56  D1 000F1      CMPL    FCB, ORG_FCB                            ; 2794
                              0D  13 000F5      BEQL    13$                                     ;
                        08    AC  DD 000F7      PUSHL   ORG_FCB                                 ; 2797
                              7E  D4 000FA      CLRL    -(SP)                                   ;
       0000G  CF              02  FB 000FC      CALLS   #2, READ_HEADER                         ;
              52              50  D0 00101      MOVL    R0, HEADER                              ;
              50        08    AC  D0 00104 13$: MOVL    ORG_FCB, R0                             ; 2802
         58   A0        04    AC  D0 00108      MOVL    UIC, 88(R0)                             ;
              50              01  D0 0010D 14$: MOVL    #1, R0                                  ; 2804
                                  04 00110      RET                                             ; 2806

; Routine Size:  273 bytes,    Routine Base:  $CODE$ + 0CED
```

```
;1825     2807  1 ROUTINE CHANGE_CLASS (HEADER, CLASS_BLOCK) : L_NORM =
;1826     2808  1
;1827     2809  1 !++
;1828     2810  1 !
;1829     2811  1 ! FUNCTIONAL DESCRIPTION:
;1830     2812  1 !
;1831     2813  1 !       This routine changes the classification of the file.  It checks to
;1832     2814  1 !       make sure that the new classification is a subset of the old (unless
;1833     2815  1 !       UPGRADE or DOWNGRADE privileges are used), and that the new
;1834     2816  1 !       classification is within the bounds of the requestors authorization.
;1835     2817  1 !
;1836     2818  1 ! CALLING SEQUENCE:
;1837     2819  1 !       CHANGE_CLASS (ARG1, ARG2)
;1838     2820  1 !
;1839     2821  1 ! INPUT PARAMETERS:
;1840     2822  1 !       ARG1: address of file header
;1841     2823  1 !       ARG2: new classification block
;1842     2824  1 !
;1843     2825  1 ! IMPLICIT INPUTS:
;1844     2826  1 !       none
;1845     2827  1 !
;1846     2828  1 ! OUTPUT PARAMETERS:
;1847     2829  1 !       NONE
;1848     2830  1 !
;1849     2831  1 ! IMPLICIT OUTPUTS:
;1850     2832  1 !       NONE
;1851     2833  1 !
;1852     2834  1 ! ROUTINE VALUE:
;1853     2835  1 !       NONE
;1854     2836  1 !
;1855     2837  1 ! SIDE EFFECTS:
;1856     2838  1 !       Primary file header modified and FCB updated with new classification.
;1857     2839  1 !
;1858     2840  1 !--
;1859     2841  1
;1860     2842  2 BEGIN
;1861     2843  2
;1862     2844  2 MAP
;1863     2845  2         HEADER          : REF BBLOCK,   ! file header arg
;1864     2846  2         CLASS_BLOCK     : REF BBLOCK;   ! Address of new classification block
;1865     2847  2
;1866     2848  2 LINKAGE
;1867     2849  2         L_CHECKCLASS    = JSB (REGISTER = 2, REGISTER = 3,
;1868     2850  2                                REGISTER = 4, REGISTER = 5, REGISTER = 6;
;1869     2851  2                                REGISTER = 1)
;1870     2852  2                         : NOTUSED (7, 8, 9, 10,11);
;1871     2853  2
;1872     2854  2 LOCAL
;1873     2855  2         STATUS,                         ! result of protection check
;1874     2856  2         PHD             : REF BBLOCK,   ! PHD address
;1875     2857  2         FCB             : REF BBLOCK;   ! FCB of current header, if any
;1876     2858  2
;1877     2859  2 BIND_COMMON;
;1878     2860  2
;1879     2861  2 EXTERNAL
;1880     2862  2         CTL$GL_PHD      : ADDRESSING_MODE (GENERAL);    ! PHD pointer
;1881     2863  2
```

```
;  1882          2864  2 EXTERNAL ROUTINE
;  1883          2865  2         LOCK_COUNT      : L_NORM,       ! determine if other locks exist
;  1884          2866  2         EXE$CHECKCLASS  : L_CHECKCLASS ADDRESSING_MODE (GENERAL);
;  1885          2867  2                                         ! Classification checking routine
;  1886          2868  2
;  1887          2869  2
;  1888          2870  2 ! Check to make sure that the file is not open. Changing the classification
;  1889          2871  2 ! of an open file (either upwards or downwards) creates a storage channel.
;  1890          2872  2 !
;  1891          2873  2
;  1892          2874  2 FCB = .PRIMARY_FCB;
;  1893          2875  2 IF .FCB[FCB$W_REFCNT] NEQ 0
;  1894          2876  2     OR LOCK_COUNT (.FCB [FCB$L_ACCLKID]) NEQ 1
;  1895          2877  2 THEN RETURN SS$_ACCONFLICT;
;  1896          2878  2
;  1897          2879  2 ! Check to see if there is a classification block in the file header.
;  1898          2880  2 !
;  1899          2881  2
;  1900          2882  2 IF .HEADER[FH2$B_IDOFFSET]
;  1901          2883  2    LEQU ($BYTEOFFSET (FH2$R_CLASS_PROT) + FH2$S_CLASS_PROT) / 2
;  1902          2884  2 THEN RETURN 1;
;  1903          2885  2
;  1904          2886  2 ! There is.  Check the old versus the new classification.
;  1905          2887  2 !
;  1906          2888  2
;  1907          2889  2 STATUS = EXE$CHECKCLASS (LOCAL_ARB[ARB$Q_PRIV], CHP$M_WRITE,
;  1908          2890  2                          HEADER[FH2$R_CLASS_PROT], .CLASS_BLOCK, .CLASS_BLOCK);
;  1909          2891  2
;  1910          2892  2 ! Now insure that the new classification is within the accessor's authorized
;  1911          2893  2 ! bounds.
;  1912          2894  2 !
;  1913          2895  2
;  1914          2896  2 PHD = .CTL$GL_PHD;
;  1915          2897  2
;  1916          2898  2 IF .STATUS THEN
;  1917          2899  2 STATUS = EXE$CHECKCLASS (LOCAL_ARB[ARB$Q_PRIV], CHP$M_READ OR CHP$M_WRITE,
;  1918          2900  2                          .CLASS_BLOCK, PHD[PHD$R_MIN_CLASS],
;  1919          2901  2                          PHD[PHD$R_MAX_CLASS]);
;  1920          2902  2
;  1921          2903  2 ! Finally check the new classification against the limits of the volume.
;  1922          2904  2 !
;  1923          2905  2
;  1924          2906  2 IF .STATUS THEN
;  1925          2907  2 STATUS = EXE$CHECKCLASS (LOCAL_ARB[ARB$Q_PRIV], CHP$M_READ OR CHP$M_WRITE,
;  1926          2908  2                          .CLASS_BLOCK, CURRENT_VCB[VCB$R_MIN_CLASS],
;  1927          2909  2                          CURRENT_VCB[VCB$R_MAX_CLASS]);
;  1928          2910  2
;  1929          2911  2 ! BYPASS privilege overrides any protection failures.
;  1930          2912  2 !
;  1931          2913  2
;  1932          2914  2 IF NOT .STATUS
;  1933          2915  2 AND NOT .BBLOCK [LOCAL_ARB[ARB$Q_PRIV], PRV$V_BYPASS]
;  1934          2916  2 THEN RETURN SS$_NOPRIV;
;  1935          2917  2
;  1936          2918  2 ! The classification check has passed.  Save the new classification in the
;  1937          2919  2 ! file header.
;  1938          2920  2 !
```

```
;  1939          2921  2
;  1940          2922  2 CH$MOVE (FH2$S_CLASS_PROT, .CLASS_BLOCK, HEADER[FH2$R_CLASS_PROT]);
;  1941          2923  2
;  1942          2924  2 ! Now move the new classification block to the FCB.
;  1943          2925  2 !
;  1944          2926  2
;  1945          2927  2 CH$MOVE (FH2$S_CLASS_PROT, .CLASS_BLOCK, FCB[FCB$R_MIN_CLASS_PROT]);
;  1946          2928  2 CH$MOVE (FH2$S_CLASS_PROT, .CLASS_BLOCK, FCB[FCB$R_MAX_CLASS_PROT]);
;  1947          2929  2
;  1948          2930  2 1
;  1949          2931  1 END;                                    ! end of routine CHANGE_CLASS


                                                        .EXTRN  CTL$GL_PHD, LOCK_COUNT
                                                        .EXTRN  EXE$CHECKCLASS

                                        01FC 00000 CHANGE_CLASS:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8                       ; 2807
                          58 00000000G   00   9E 00002          MOVAB   EXE$CHECKCLASS, R8                      ;
                          52     0284    CA   9E 00009          MOVAB   644(BASE), R2                           ; 2857
                          57       08    AA   D0 0000F          MOVL    8(BASE), FCB                            ; 2874
                                   18    A7   B5 00012          TSTW    24(FCB)                                 ; 2875
                                         0D   12 00015          BNEQ    1$                                      ;
                                   48    A7   DD 00017          PUSHL   72(FCB)                                 ; 2876
                    0000G  CF              01   FB 0001A          CALLS   #1, LOCK_COUNT                          ;
                          01               50   D1 0001F          CMPL    R0, #1                                  ;
                                         06   13 00022          BEQL    2$                                      ;
                          50     0800    8F   3C 00024 1$:       MOVZWL  #2048, R0                               ; 2877
                                         04   00029              RET                                             ;
                          36       04    BC   91 0002A 2$:       CMPB    @HEADER, #54                            ; 2883
                                         74   1B 0002E          BLEQU   5$                                      ;
                    54          04    AC 00000058   8F   C1 00030          ADDL3   #88, HEADER, R4                 ; 2890
                          56       08    AC   D0 00039          MOVL    CLASS_BLOCK, R6                         ;
                          55       08    AC   D0 0003D          MOVL    CLASS_BLOCK, R5                         ;
                          53             02   D0 00041          MOVL    #2, R3                                  ;
                                         68   16 00044          JSB     EXE$CHECKCLASS                          ;
                          51 00000000G   00   D0 00046          MOVL    CTL$GL_PHD, PHD                         ; 2896
                          34             50   E9 0004D          BLBC    STATUS, 3$                              ; 2898
                          56     0128    C1   9E 00050          MOVAB   296(PHD), R6                            ; 2901
                          55     0114    C1   9E 00055          MOVAB   276(PHD), R5                            ; 2900
                          54       08    AC   D0 0005A          MOVL    CLASS_BLOCK, R4                         ; 2901
                          53             03   D0 0005E          MOVL    #3, R3                                  ;
                                         68   16 00061          JSB     EXE$CHECKCLASS                          ;
                                   1E    50   E9 00063          BLBC    STATUS, 3$                              ; 2906
                    56          98    AA 000000D8   8F   C1 00066          ADDL3   #216, -104(BASE), R6            ; 2909
                    55          98    AA 000000C4   8F   C1 0006F          ADDL3   #196, -104(BASE), R5            ; 2908
                          54       08    AC   D0 00078          MOVL    CLASS_BLOCK, R4                         ; 2909
                          53             03   D0 0007C          MOVL    #3, R3                                  ;
                                         68   16 0007F          JSB     EXE$CHECKCLASS                          ;
                                   08    50   E8 00081          BLBS    STATUS, 4$                              ; 2914
                    04                   62   1D E0 00084 3$:       BBS     #29, (R2), 4$                           ; 2915
                          50             24   D0 00088          MOVL    #36, R0                                 ; 2916
                                         04   00088              RET                                             ;
                          50       04    AC   D0 0008C 4$:       MOVL    HEADER, R0                              ; 2922
                 58       A0    08    BC   14 28 00090          MOVC3   #20, @CLASS_BLOCK, 88(R0)               ;
              0088       C7    08    BC   14 28 00096          MOVC3   #20, @CLASS_BLOCK, 136(FCB)             ; 2927
```

```
                009C   C7      08   BC            14  28 00009D          MOVC3   #20, @CLASS_BLOCK, 156(FCB)                ; 2928
                                    50            01  D0 000A4 5$:      MOVL    #1, R0                                     ; 2931
                                                      04 000A7          RET                                                ;

; Routine Size:  168 bytes,    Routine Base:  $CODE$ + 0DFE

; 1950          2932  1
; 1951          2933  1 END
; 1952          2934  0 ELUDOM


;                                   PSECT SUMMARY

;      Name                     Bytes                          Attributes

;  $CODE$                        3750  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                Library Statistics

;                                          -------- Symbols --------      Pages       Processing
;      File                                Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619       181         0        1000        00:02.0


;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RWATTR/OBJ=OBJ$:RWATTR MSRC$:RWATTR/UPDATE=(ENH$:RWATTR)

; Size:          3361 code + 389 data bytes
; Run Time:         01:52.8
; Elapsed Time:     03:28.7
; Lines/CPU Min:     1560
; Lexemes/CPU-Min: 39766
; Memory Used:  536 pages
; Compilation Complete
```

RWVB
LIS

RDBLOK
LIS

REQUEU
LIS

RWATTR
LIS

REMOVE
LIS

RDHEDR
LIS

RETDIR
LIS